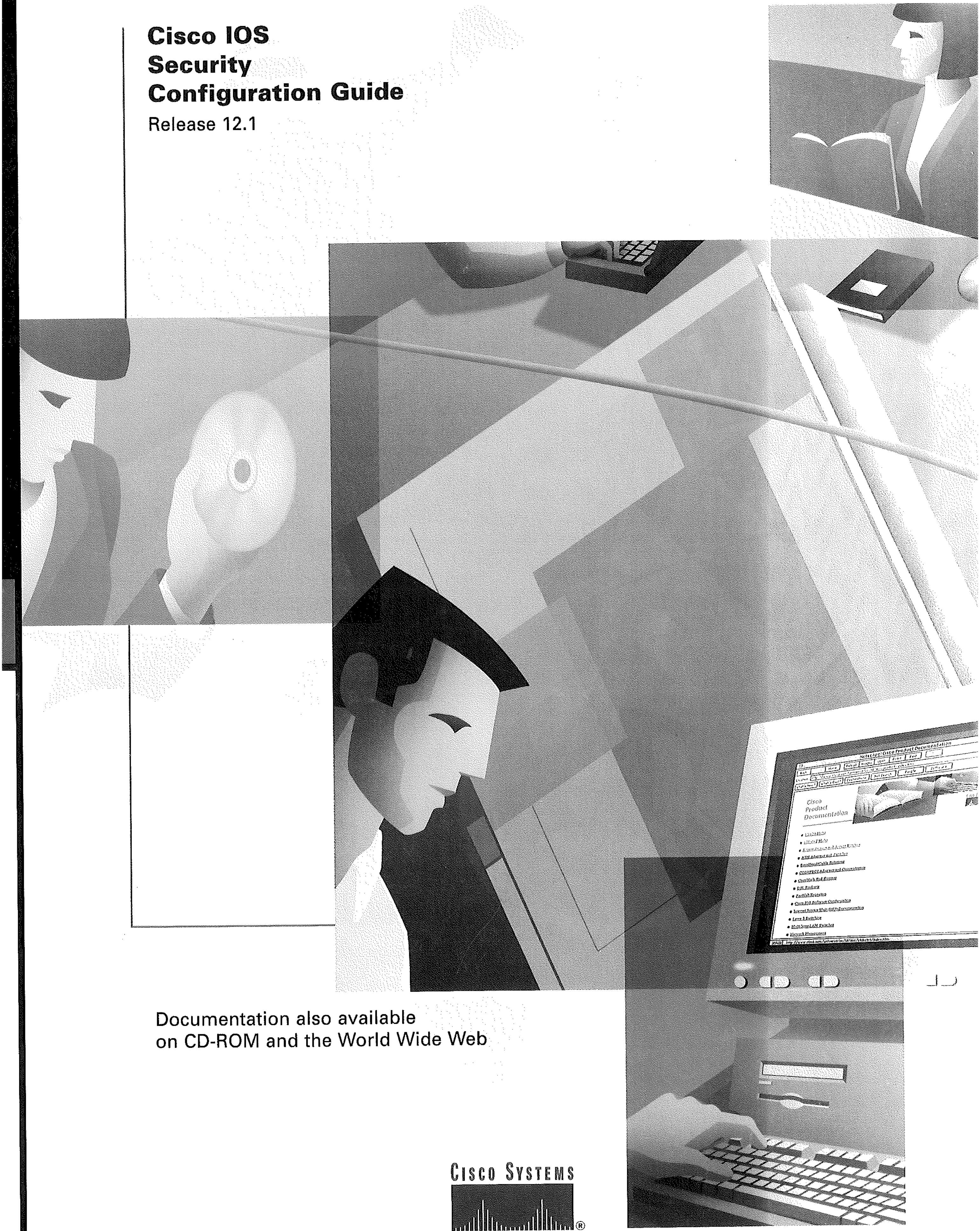
Cisco IOS Security Configuration Guide

Release 12.1



Documentation also available on CD-ROM and the World Wide Web

Cisco Systems

Liz Patton

Cisco Reader Comment Card

General Information

1 Years of networking experience ______ Years of experience with Cisco products ______

2 I have these network types: □ LAN □ Backbone □ WAN
□ Other: ______

3 I have these Cisco products: □ Switches □ Routers
□ Other: Specify model(s) ______

4 I perform these types of tasks: □ H/W Install and/or Maintenance □ S/W Config
□ Network Management □ Other: ______

5 I use these types of documentation: □ H/W Install □ H/W Config □ S/W Config
□ Command Reference □ Quick Reference □ Release Notes □ Online Help
□ Other: ______

6 I access this information through: ____% Cisco Connection Online (CCO) ____% CD-ROM
____% Printed docs ____% Other: ______

7 Which method do you prefer? ______

8 I use the following three product features the most: ______

Document-specific Information

Document Title: Cisco IOS Security Configuration Guide

Part Number: 78-10248-01 S/W Release (if applicable): 12.1

On a scale of 1–5 (5 being the best) please let us know how we rate in the following areas:

_____ The document was written at my technical level of understanding.

_____ The document was complete.

_____ The information was well organized.

_____ The information was accurate.

_____ The information I wanted was easy to find.

_____ The information I found was useful to my job.

Please comment on our lowest score(s): ______

Mailing Information

Company Name ______ Date ______

Contact Name ______ Job Title ______

Mailing Address ______

City ______ State/Province ______ ZIP/Postal Code ______

Country ______ Phone (   ) ______ Extension ______

Fax (   ) ______ E-mail ______

Can we contact you further concerning our documentation? □ Yes □ No

You can also send us your comments by e-mail to **bug-doc@cisco.com**, or fax your comments to us at **(408) 527-8089**.

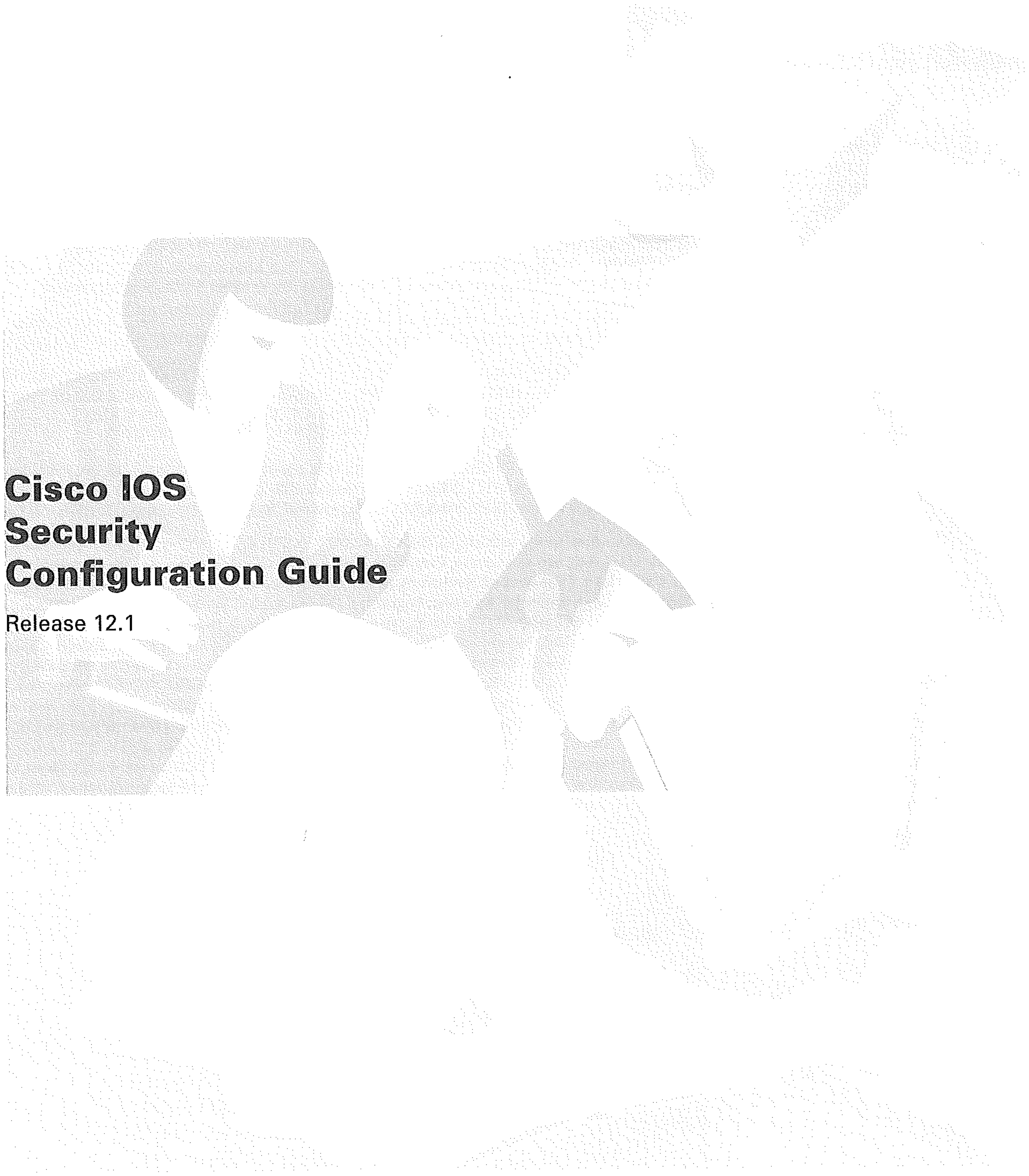
NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 4631 SAN JOSE CA

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN DOCUMENT RESOURCE CONNECTION
CISCO SYSTEMS INC
170 WEST TASMAN DRIVE
SAN JOSE CA 95134-9883



Cisco IOS Security Configuration Guide

Release 12.1

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

Customer Order Number: DOC-7810248=
Text Part Number: 78-10248-01

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Access Registrar, AccessPath, Any to Any, AtmDirector, Browse with Me, CCDA, CCDE, CCDP, CCIE, CCNA, CCNP, CCSI, CD-PAC, the Cisco logo, Cisco Certified Internetwork Expert logo, *CiscoLink*, the Cisco Management Connection logo, the Cisco Net*Works* logo, the Cisco Powered Network logo, Cisco Systems Capital, the Cisco Systems Capital logo, Cisco Systems Networking Academy, the Cisco Systems Networking Academy logo, the Cisco Technologies logo, ConnectWay, Fast Step, FireRunner, Follow Me Browsing, FormShare, GigaStack, IGX, Intelligence in the Optical Core, Internet Quotient, IP/VC, Kernel Proxy, MGX, MultiPath Data, MultiPath Voice, Natural Network Viewer, NetSonar, Network Registrar, the Networkers logo, *Packet*, PIX, Point and Click Internetworking, Policy Builder, Precept, ScriptShare, Secure Script, ServiceWay, Shop with Me, SlideCast, SMARTnet, SVX, *The Cell*, TrafficDirector, TransPath, ViewRunner, Virtual Loop Carrier System, Virtual Service Node, Virtual Voice Line, VisionWay, VlanDirector, Voice LAN, WaRP, Wavelength Router, Wavelength Router Protocol, WebViewer, Workgroup Director, and Workgroup Stack are trademarks; Changing the Way We Work, Live, Play, and Learn, Empowering the Internet Generation, The Internet Economy, and The New Internet Economy are service marks; and ASIST, BPX, Catalyst, Cisco, Cisco IOS, the Cisco IOS logo, Cisco Systems, the Cisco Systems logo, the Cisco Systems Cisco Press logo, Enterprise/Solver, EtherChannel, EtherSwitch, FastHub, FastLink, FastPAD, FastSwitch, GeoTel, IOS, IP/TV, IPX, LightStream, LightSwitch, MICA, NetRanger, Post-Routing, Pre-Routing, Registrar, StrataView Plus, Stratm, TeleRouter, and VCO are registered trademarks of Cisco Systems, Inc. or its affiliates in the U.S. and certain other countries. All other trademarks mentioned in this document are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any of its resellers. (9912R)

Cisco IOS Security Configuration Guide

Authentication, Authorization, and Accounting (AAA)

Security Server Protocols

Traffic Filtering and Firewalls

Configuring Context-Based Access Control SC-197

IP Security and Encryption

Other Security Features

Appendixes

Index

About the Cisco IOS Software Documentation

This chapter discusses the objectives, audience, organization, and conventions of the Cisco IOS software documentation. It also discusses how to obtain documentation on Cisco Connection Online and the Documentation CD-ROM.

Documentation Objectives

This Cisco IOS software documentation describes the tasks and commands necessary to configure and maintain your networking device.

Audience

The Cisco IOS software documentation is intended primarily for users who configure and maintain networking devices, but are not necessarily familiar with tasks, the relationship between tasks, or the commands necessary to perform particular tasks.

Documentation Organization

The Cisco IOS software documentation is divided into 12 modules and 2 master indexes. In addition to the main documentation set, there are 4 supporting documents.

Documentation Modules

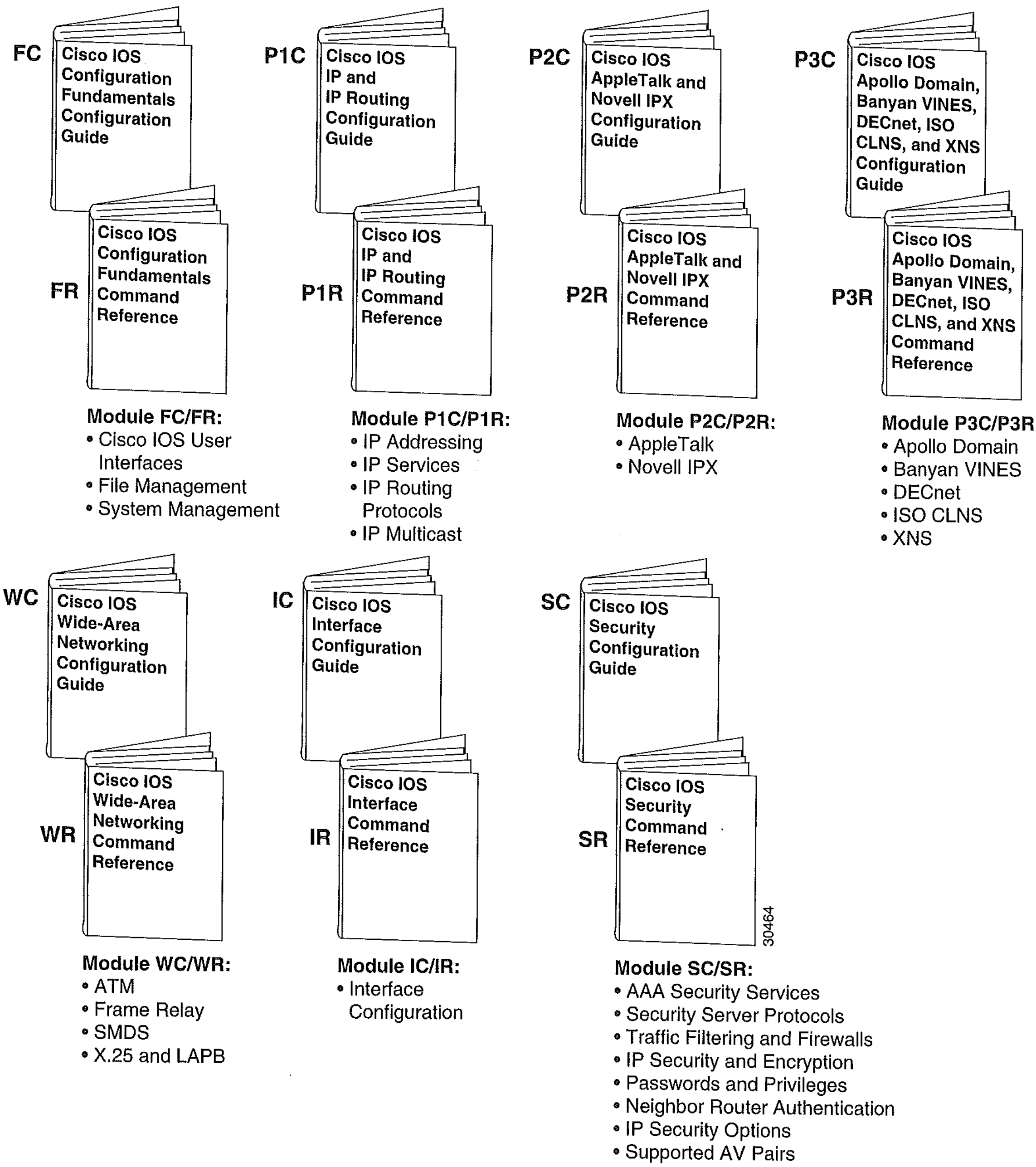
The Cisco IOS documentation modules consist of configuration guides and corresponding command reference publications. Chapters in a configuration guide describe protocols, configuration tasks, and Cisco IOS software functionality and contain comprehensive configuration examples. Chapters in a command reference publication provide complete command syntax information. Use each configuration guide in conjunction with its corresponding command reference publication.

Documentation Set

The Cisco IOS software documentation set is shown in Figure 1.

Note The abbreviations next to the book icons are page designators (for example, FC, FR, and so on), which are defined in a key in the index of each document to help with navigation. The bulleted lists under each module describe the major technology areas discussed in their corresponding books.

Figure 1 **Cisco IOS Software Documentation Modules**


FC
Cisco IOS Configuration Fundamentals Configuration Guide
FR
Cisco IOS Configuration Fundamentals Command Reference
P1C
Cisco IOS IP and IP Routing Configuration Guide
P1R
Cisco IOS IP and IP Routing Command Reference
P2C
Cisco IOS AppleTalk and Novell IPX Configuration Guide
P2R
Cisco IOS AppleTalk and Novell IPX Command Reference
P3C
Cisco IOS Apollo Domain, Banyan VINES, DECnet, ISO CLNS, and XNS Configuration Guide
P3R
Cisco IOS Apollo Domain, Banyan VINES, DECnet, ISO CLNS, and XNS Command Reference
WC
Cisco IOS Wide-Area Networking Configuration Guide
WR
Cisco IOS Wide-Area Networking Command Reference
IC
Cisco IOS Interface Configuration Guide
IR
Cisco IOS Interface Command Reference
SC
Cisco IOS Security Configuration Guide
SR
Cisco IOS Security Command Reference
30464

Module FC/FR:
- Cisco IOS User Interfaces
- File Management
- System Management

Module P1C/P1R:
- IP Addressing
- IP Services
- IP Routing Protocols
- IP Multicast

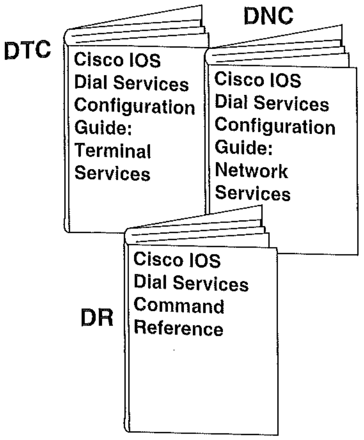
Module P2C/P2R:
- AppleTalk
- Novell IPX

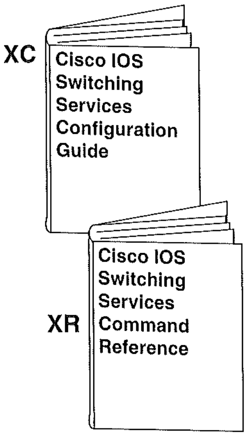
Module P3C/P3R:
- Apollo Domain
- Banyan VINES
- DECnet
- ISO CLNS
- XNS

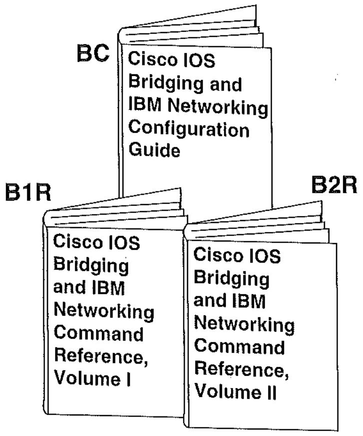
Module WC/WR:
- ATM
- Frame Relay
- SMDS
- X.25 and LAPB

Module IC/IR:
- Interface Configuration

Module SC/SR:
- AAA Security Services
- Security Server Protocols
- Traffic Filtering and Firewalls
- IP Security and Encryption
- Passwords and Privileges
- Neighbor Router Authentication
- IP Security Options
- Supported AV Pairs


DTC
Cisco IOS Dial Services Configuration Guide: Terminal Services
DNC
Cisco IOS Dial Services Configuration Guide: Network Services
DR
Cisco IOS Dial Services Command Reference


XC
Cisco IOS Switching Services Configuration Guide
XR
Cisco IOS Switching Services Command Reference


BC
Cisco IOS Bridging and IBM Networking Configuration Guide
B1R
Cisco IOS Bridging and IBM Networking Command Reference, Volume I
B2R
Cisco IOS Bridging and IBM Networking Command Reference, Volume II

Module DTC/DR:

- Dial Access
- Modem Management
- ISDN BRI Services
- Point-to-Point Protocols
- Dial-on-Demand Routing
- Dial Backup
- Terminal Services

Module DNC/DR:

- Large-Scale Dial Solutions
- Cost-Control Solutions
- Virtual Private Networks
- X.25 on ISDN Solutions
- Telco Solutions
- Dial-Related Addressing Services
- Internetworking Dial Access Scenarios

Module XC/XR:

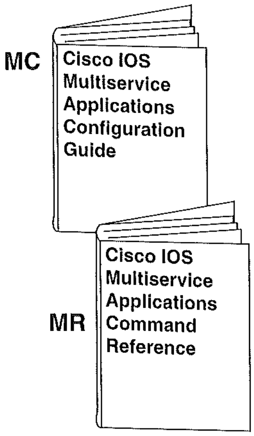
- Cisco IOS Switching Paths
- Cisco Express Forwarding
- NetFlow Switching
- Multiprotocol Label Switching
- Multilayer Switching
- Multicast Distributed Switching
- Virtual LANs
- LAN Emulation

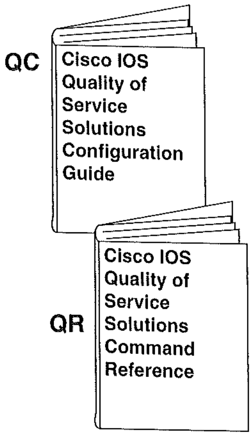
Module BC/B1R:

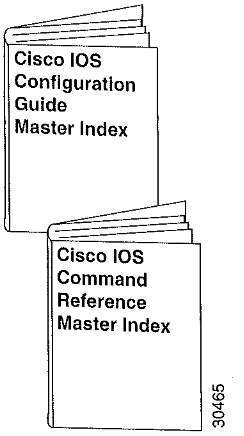
- Transparent Bridging
- Source-Route Bridging
- Token Ring Inter-Switch Link
- Token Ring Route Switch Module
- Remote Source-Route Bridging
- Data-Link Switching Plus
- Serial Tunnel and Block Serial Tunnel
- LLC2 and SDLC
- IBM Network Media Translation
- SNA Frame Relay Access
- NCIA Client/Server
- Airline Product Set

Module BC/B2R:

- DSPU and SNA Service Point
- SNA Switching Services
- Cisco Transaction Connection
- Cisco Mainframe Channel Connection
- CLAW and TCP/IP Offload
- CSNA, CMPC, and CMPC+
- TN3270 Server


MC
Cisco IOS Multiservice Applications Configuration Guide
MR
Cisco IOS Multiservice Applications Command Reference


QC
Cisco IOS Quality of Service Solutions Configuration Guide
QR
Cisco IOS Quality of Service Solutions Command Reference


Cisco IOS Configuration Guide Master Index
Cisco IOS Command Reference Master Index
30465

Module MC/MR:

- Voice over IP
- Voice over Frame Relay
- Voice over ATM
- Voice over HDLC
- Video Support
- Universal Broadband Features

Module QC/QR:

- Packet Classification
- Congestion Management
- Congestion Avoidance
- Policing and Shaping
- Signalling
- Link Efficiency Mechanisms

Master Indexes

Two master indexes provide indexing information for the Cisco IOS software documentation set: an index for the configuration guides, and an index for the command references. In addition, individual books contain a book-specific index.

Supporting Documents

The following documents support the Cisco IOS software documentation set:

- *Cisco IOS Command Summary*
- *Cisco IOS System Error Messages*
- *Cisco IOS Debug Command Reference*
- *Cisco IOS Dial Services Quick Configuration Guide*

New and Changed Information

The following is new or changed information since the last release of the *Cisco IOS Security Configuration Guide*:

- The TACACS and Extended TACACS security protocols are no longer supported in this release. References to these protocols have been removed throughout the book. For AAA server host information, refer to the "Security Server Protocols" section, which describes RADIUS, TACACS+, and Kerberos support.
- The Cisco IOS Firewall Feature Set is now known as Cisco Secure Integrated Software. For a detailed description of Cisco Secure Integrated Software features, refer to the "Traffic Filtering and Firewalls" section.
- The "Traffic Filtering and Firewalls" section has three new chapters, covering three new features: Integrated Intrusion Detection System, Authentication Proxy, and Port to Application Mapping.Cisco IOS Security Configuration Guide

Document Conventions

The Cisco IOS documentation set uses the following conventions:

Convention	**Description**
^ or Ctrl	^ or Ctrl represents the Control key. For example, the key combination ^D or Ctrl-D means hold down the Control key while you press the D key. Keys are indicated in capital letters but are not case sensitive.
string	A string is a nonquoted set of characters. For example, when setting an SNMP community string to public, do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

Convention	Description
`screen`	Courier plain shows an example of information displayed on the screen.
`boldface screen`	Courier bold shows an example of text that you must enter.
< >	Angle brackets show nonprinting characters, such as passwords.
!	An exclamation point at the beginning of a line indicates a comment line. (Exclamation points are also displayed by the Cisco IOS software for certain processes.)
[ ]	Square brackets show default responses to system prompts.

The following conventions are used to attract the attention of the reader:

⚠ **Caution** Means *reader be careful.* In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note.* Notes contain helpful suggestions or references to materials not contained in this manual.

Timesaver Means the *described action saves time.* You can save time by performing the action described in the paragraph.

Within the Cisco IOS software documentation, the term *router* is generally used to refer to a variety of networking devices (for example, routers, access servers, and Route Switch Modules). Within examples, routers, access servers, and other networking devices that support Cisco IOS software are shown alternately. These products are used only for example purposes; that is, an example that shows one product does not indicate that other products are not supported.

Command Syntax Conventions

Command descriptions use the following conventions:

Convention	Description
boldface	Boldface text indicates commands and keywords that you enter literally as shown.
italics	Italic text indicates arguments for which you supply values.
[x]	Square brackets indicate an optional element (keyword or argument).

{x} — Braces indicate a required element (keyword or argument).

[x {y | z}] — Braces and vertical lines within square brackets indicate a required choice within an optional element.

Cisco Connection Online

Cisco Connection Online (CCO) is the primary, real-time support channel for Cisco Systems. Maintenance customers and partners can self-register on CCO to obtain additional information and services.

Available 24 hours a day, 7 days a week, CCO provides a wealth of standard and value-added services to customers and business partners of Cisco Systems. CCO services include product information, product documentation, software updates, release notes, technical tips, the Bug Navigator, configuration notes, brochures, descriptions of service offerings, and download access to public and authorized files.

CCO serves a wide variety of users through two interfaces that are updated and enhanced simultaneously: a character-based version and a multimedia version that resides on the World Wide Web (WWW). The character-based CCO supports Zmodem, Kermit, Xmodem, FTP, and Internet e-mail, and it is excellent for quick access to information over lower bandwidths. The WWW version of CCO provides richly formatted documents with photographs, figures, graphics, and video, as well as hyperlinks to related information.

You can access CCO in the following ways:

- http://www.cisco.com
- http://www-europe.cisco.com
- http://www-china.cisco.com
- Telnet: cco.cisco.com
- Modem: From North America, 408 526-8070; from Europe, 33 1 64 46 40 82. Use the following terminal settings: VT100 emulation; databits: 8; parity: none; stop bits: 1; and connection rates up to 28.8 kbps.

For a copy of the CCO Frequently Asked Questions (FAQ), contact cco-help@cisco.com. For additional information, contact cco-team@cisco.com.

Note If you are a network administrator and need personal technical assistance with a Cisco product that is under warranty or covered by a maintenance contract, contact the Cisco Technical Assistance Center (TAC) at 800 553-2447, 408 526-7209, or tac@cisco.com. To obtain general information about Cisco Systems, Cisco products, or upgrades, contact 800 553-6387, 408 526-7208, or cs-rep@cisco.com.

Documentation CD-ROM

Cisco documentation and additional literature are available in a CD-ROM package, which ships with your product. The Documentation CD-ROM, a member of the Cisco Connection Family, is updated monthly; therefore, it might be more current than printed documentation. To order additional copies of the Documentation CD-ROM, contact your local sales representative or call customer service. The

CD-ROM package is available as a single package or as an annual subscription. You can also access Cisco documentation on the World Wide Web at http://www.cisco.com, http://www-china.cisco.com, or http://www-europe.cisco.com.

Providing Documentation Feedback

If you are reading Cisco product documentation on the World Wide Web, you can submit comments electronically. Click **Feedback** in the toolbar and select **Documentation**. After you complete the form, click **Submit** to send it to Cisco.

You can also submit feedback on Cisco documentation as follows:

- Mail in the Cisco Reader Comment Card located at the front of this book
- Send an e-mail to bug-doc@cisco.com
- Send a fax to 408 527-8089

We appreciate your comments.

Using Cisco IOS Software

This chapter provides helpful tips for understanding and configuring Cisco IOS software using the command-line interface (CLI). It contains the following sections:

- Understanding Command Modes
- Getting Help
- Using the No and Default Forms of Commands
- Saving Configuration Changes
- Searching and Filtering Output of show and more Commands

For an overview of Cisco IOS software configuration, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide.*

For information on the conventions used in the Cisco IOS documentation set, see the "About the Cisco IOS Software Documentation" chapter located at the beginning of this book.

Understanding Command Modes

The Cisco IOS user interface is divided into many different modes. The commands available to you at any given time depend on which mode you are currently in. Entering a question mark (**?**) at the system prompt allows you to obtain a list of commands available for each command mode.

When you log in to the Cisco IOS software, you begin in user mode, often called EXEC mode. Only a limited subset of the commands are available in EXEC mode. To have access to all commands, you must enter privileged EXEC mode. Normally, you must enter a password to enter privileged EXEC mode. From privileged mode, you can enter any EXEC command or enter global configuration mode. Most of the EXEC commands are one-time commands, such as **show** commands, which show important status information, and **clear** commands, which clear counters or interfaces. The EXEC commands are not saved when the networking device reboots.

The configuration modes allow you to make changes to the running configuration. If you later save the configuration to the startup configuration, these commands are stored when the networking device reboots. To enter the various configuration modes, you must start at global configuration mode. From global configuration mode, you can enter interface configuration mode, subinterface configuration mode, and a variety of protocol-specific modes.

ROM monitor mode is a separate mode used when a networking device running Cisco IOS software cannot boot properly. If your networking device does not find a valid system image when it is booting, or if its configuration file is corrupted at startup, the system might enter ROM monitor mode.

Summary of Main Command Modes

Table 1 summarizes the main command modes of the Cisco IOS software.

Table 1 **Summary of Main Command Modes**

Command Mode	Access Method	Prompt	Exit Method
User EXEC	Log in.	`Router>`	Use the **logout** command.
Privileged EXEC	From user EXEC mode, use the **enable** EXEC command.	`Router#`	To exit back to user EXEC mode, use the **disable** command. To enter global configuration mode, use the **configure terminal** privileged EXEC command.
Global configuration	From privileged EXEC mode, use the **configure terminal** privileged EXEC command.	`Router(config)#`	To exit to privileged EXEC mode, use the **exit** or **end** command or press **Ctrl-Z**. To enter interface configuration mode, use an **interface** configuration command.
Interface configuration	From global configuration mode, enter by specifying an interface with an **interface** command.	`Router(config-if)#`	To exit to global configuration mode, use the **exit** command. To exit to privileged EXEC mode, use the **exit** command or press **Ctrl-Z**. To enter subinterface configuration mode, specify a subinterface with the **interface** command.
Subinterface configuration	From interface configuration mode, specify a subinterface with an **interface** command.	`Router(config-subif)#`	To exit to global configuration mode, use the **exit** command. To enter privileged EXEC mode, use the **end** command or press **Ctrl-Z**.
ROM monitor	From privileged EXEC mode, use the **reload** EXEC command. Press the Break key during the first 60 seconds while the system is booting.	`>`	To exit to user EXEC mode, use the **continue** command.

For more information regarding command modes, refer to the "Using the Command-Line Interface" chapter in the *Cisco IOS Configuration Fundamentals Configuration Guide.*

Getting Help

Entering a question mark (?) at the system prompt displays a list of commands available for each command mode. You can also get a list of keywords and arguments associated with any command by using the context-sensitive help feature.

To get help specific to a command mode, a command, a keyword, or an argument, use one of the following commands:

Command	Purpose
help	Obtains a brief description of the help system in any command mode.
*abbreviated-command-entry***?**	Obtains a list of commands that begin with a particular character string. (No space between command and question mark.)
abbreviated-command-entry<**Tab**>	Completes a partial command name.
?	Lists all commands available for a particular command mode.
command **?**	Lists the keywords or arguments that you must enter next on the command line. (Space between command and question mark.)

Example: How to Find Command Options

This section provides an example of how to display syntax for a command. The syntax can consist of optional or required keywords and arguments. To display keywords and arguments for a command, enter a question mark (**?**) at the configuration prompt, or after entering part of a command followed by a space. The Cisco IOS software displays a list of keywords and arguments available along with a brief description of them. For example, if you were in global configuration mode, typed the command **arap**, and wanted to see all the keywords or arguments that may be entered next on the command line, you would type **arap ?**.

Table 2 shows examples of how you can use the question mark (**?**) to assist you in entering commands. The table steps you through configuring a serial interface IP address on a Cisco 7206 router running Cisco IOS Release 12.0(3).

Table 2 How to Find Command Options

Command	Comment

`Router> enable`
`Password: <password>`
`Router#`
	Enter the **enable** command and password to access privileged EXEC commands.
	You are in privileged EXEC mode when the prompt changes to `Router#`.

`Router# configure terminal`
`Enter configuration commands, one per line. End with CNTL/Z.`
`Router(config)#`
	Enter the **configure terminal** privileged EXEC command to enter global configuration mode.
	You are in global configuration mode when the prompt changes to `Router(config)#`.

`Router(config)# interface serial ?`
`  <0-6>      Serial interface number`
`Router(config)# interface serial 4 ?`
`  /`
`Router(config)# interface serial 4/ ?`
`  <0-3>      Serial interface number`
`Router(config)# interface serial 4/0`
`Router(config-if)#`
	Enter interface configuration mode by specifying the serial interface that you want to configure using the **interface serial** global configuration command.
	Enter a **?** to display what you must enter next on the command line. In this example, you must enter the serial interface slot number and port number, separated by a back slash.
	You are in interface configuration mode when the prompt changes to `Router(config-if)#`.

Table 2 How to Find Command Options (continued)

<table>
<tr><th>Command</th><th>Comment</th></tr>
<tr><td>

```
Router(config-if)# ?
Interface configuration commands:
  ...
  ip                    Interface Internet Protocol config commands
  keepalive             Enable keepalive
  lan-name              LAN Name command
  llc2                  LLC2 Interface Subcommands
  load-interval         Specify interval for load calculation for an
                        interface
  locaddr-priority      Assign a priority group
  logging               Configure logging for interface
  loopback              Configure internal loopback on an interface
  mac-address           Manually set interface MAC address
  mls                   mls router sub/interface commands
  mpoa                  MPOA interface configuration commands
  mtu                   Set the interface Maximum Transmission Unit (MTU)
  netbios               Use a defined NETBIOS access list or enable
                        name-caching
  no                    Negate a command or set its defaults
  nrzi-encoding         Enable use of NRZI encoding
  ntp                   Configure NTP
  ...
Router(config-if)#
```

</td><td>

Enter a **?** to display a list of all the interface configuration commands available for the serial interface. This example shows only some of the interface configuration commands that are available.

</td></tr>
<tr><td>

```
Router(config-if)# ip ?
Interface IP configuration subcommands:
  access-group          Specify access control for packets
  accounting            Enable IP accounting on this interface
  address               Set the IP address of an interface
  authentication        authentication subcommands
  bandwidth-percent     Set EIGRP bandwidth limit
  broadcast-address     Set the broadcast address of an interface
  cgmp                  Enable/disable CGMP
  directed-broadcast    Enable forwarding of directed broadcasts
  dvmrp                 DVMRP interface commands
  hello-interval        Configures IP-EIGRP hello interval
  helper-address        Specify a destination address for UDP broadcasts
  hold-time             Configures IP-EIGRP hold time
  ...
Router(config-if)# ip
```

</td><td>

Enter the command that you want to configure for the interface. In this example, the **ip** command is used.

Enter a **?** to display what you must enter next on the command line. This example shows only some of the interface IP configuration subcommands that are available.

</td></tr>
<tr><td>

```
Router(config-if)# ip address ?
  A.B.C.D               IP address
  negotiated            IP Address negotiated over PPP
Router(config-if)# ip address
```

</td><td>

Enter the subcommand that you want to configure for the interface. In this example, the **address** subcommand is entered.

Enter a **?** to display what you must enter next on the command line. In this example, you must enter an IP address or the **negotiated** keyword.

Because a carriage return (<cr>) is not displayed, it indicates that you must enter more keywords or arguments to complete the command.

</td></tr>
</table>

Table 2 How to Find Command Options (continued)

Command	Comment

`Router(config-if)# ip address 172.16.0.1 ?`
`  A.B.C.D            IP subnet mask`
`Router(config-if)# ip address 172.16.0.1`
	Enter the keyword or argument you want to use. In this example, the 172.16.0.1 IP address is entered.
	Enter a **?** to display what you must enter next on the command line. In this example, you must enter an IP subnet mask.
	Because a <cr> is not displayed, it indicates that you must enter more keywords or arguments to complete the command.

`Router(config-if)# ip address 172.16.0.1 255.255.255.0 ?`
`  secondary          Make this IP address a secondary address`
`  <cr>`
`Router(config-if)# ip address 172.16.0.1 255.255.255.0`
	Enter the IP subnet mask. In this example, the 255.255.255.0 IP subnet mask is entered.
	Enter a **?** to display what you must enter next on the command line. In this example, you can enter the **secondary** keyword or press **Enter**.
	Because a <cr> is displayed, it indicates that you can press **Enter** to complete the command.

`Router(config-if)# ip address 172.16.0.1 255.255.255.0`
`Router(config-if)#`
	In this example, **Enter** is pressed to complete the command.

Using the No and Default Forms of Commands

Almost every configuration command also has a **no** form. In general, use the **no** form to disable a function. Use the command without the keyword **no** to reenable a disabled function or to enable a function that is disabled by default. For example, IP routing is enabled by default. To disable IP routing, use the **no ip routing** command and specify **ip routing** to reenable it. The Cisco IOS software command reference publications provide the complete syntax for the configuration commands and describe what the **no** form of a command does.

Configuration commands can also have a **default** form. The **default** form of a command returns the command setting to its default. Most commands are disabled by default, so the **default** form is the same as the **no** form. However, some commands are enabled by default and have variables set to certain default values. In these cases, the **default** command enables the command and sets variables to their default values. The Cisco IOS software command reference publications describe what the **default** form of a command does if the command is not the same as the **no** form.

Saving Configuration Changes

Enter the **copy system:running-config nvram:startup-config** command to save your configuration changes to your startup configuration so that they will not be lost if there is a system reload or power outage. For example:

```
Router# copy system:running-config nvram:startup-config
Building configuration...
```

It might take a minute or two to save the configuration. After the configuration has been saved, the following output appears:

```
[OK]
Router#
```

On most platforms, this task saves the configuration to NVRAM. On the Class A Flash file system platforms, this task saves the configuration to the location specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM.

Searching and Filtering Output of show and more Commands

In Cisco IOS Release 12.0(1)T or later, you can search and filter the output for **show** and **more** commands. This functionality is useful when you need to sort through large amounts of output, or if you want to exclude output that you do not need to see.

To use this functionality, enter a **show** or **more** command followed by the "pipe" character (|), one of the keywords **begin**, **include**, or **exclude**, and an expression that you want to search or filter on:

command | {**begin** | **include** | **exclude**} *regular-expression*

The following is an example of the **show interface** command in which you want the output to only include lines where the expression "protocol" appears:

```
Router# show interface | include protocol
FastEthernet0/0 is up, line protocol is up
Serial4/0 is up, line protocol is up
Serial4/1 is up, line protocol is up
Serial4/2 is administratively down, line protocol is down
Serial4/3 is administratively down, line protocol is down
```

For more information on the search and filter functionality, refer to the "Using the Command-Line Interface" chapter in the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Security Overview

This chapter contains the following sections:

- About This Guide

  Preview the topics in this guide.

- Creating Effective Security Policies

  Learn tips and hints for creating a security policy for your organization. A security policy should be finalized and up to date *before* you configure any security features.

- Identifying Security Risks and Cisco IOS Solutions

  Identify common security risks that might be present in your network, and find the right Cisco IOS security feature to prevent security break-ins.

About This Guide

The *Cisco IOS Security Configuration Guide* describes how to configure Cisco IOS security features for your Cisco networking devices. These security features can protect your network against degradation or failure, and data loss or compromise, resulting from intentional attacks or from unintended but damaging mistakes by well-meaning network users.

This guide is divided into five parts:

- Authentication, Authorization, and Accounting (AAA)
- Security Server Protocols
- Traffic Filtering and Firewalls
- IP Security and Encryption
- Other Security Features
- Appendixes

The following sections briefly describe each of these parts.

Authentication, Authorization, and Accounting (AAA)

This part describes how to configure Cisco's authentication, authorization, and accounting (AAA) paradigm. AAA is an architectural framework for configuring a set of three independent security functions in a consistent, modular manner.

- Authentication—Provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol you select, encryption. Authentication is the way a user is identified prior to being allowed access to the network and network services. You configure AAA authentication by defining a named list of authentication methods, and then applying that list to various interfaces.
- Authorization—Provides the method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, ARA, and Telnet.

  Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights, with the appropriate user. AAA authorization works by assembling a set of attributes that describe what the user is authorized to perform. These attributes are compared to the information contained in a database for a given user and the result is returned to AAA to determine the user's actual capabilities and restrictions.
- Accounting—Provides the method for collecting and sending security server information used for billing, auditing, and reporting, such as user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes. Accounting enables you to track the services users are accessing as well as the amount of network resources they are consuming.

Note You can configure authentication outside of AAA. However, you must configure AAA if you want to use RADIUS, TACACS+, or Kerberos or if you want to configure a backup authentication method.

Security Server Protocols

In many circumstances, AAA uses security protocols to administer its security functions. If your router or access server is acting as a network access server, AAA is the means through which you establish communication between your network access server and your RADIUS, TACACS+, or Kerberos security server.

The chapters in this part describe how to configure the following security server protocols:

- RADIUS—A distributed client/server system implemented through AAA that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.
- TACACS+—A security application implemented through AAA that provides centralized validation of users attempting to gain access to a router or network access server. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation. TACACS+ provides for separate and modular authentication, authorization, and accounting facilities.
- Kerberos—A secret-key network authentication protocol implemented through AAA that uses the Data Encryption Standard (DES) cryptographic algorithm for encryption and authentication. Kerberos was designed to authenticate requests for network resources. Kerberos is based on the concept of a trusted third party that performs secure verification of users and services. The

primary use of Kerberos is to verify that users and the network services they use are really who and what they claim to be. To accomplish this, a trusted Kerberos server issues tickets to users. These tickets, which have a limited lifespan, are stored in a user's credential cache and can be used in place of the standard username-and-password authentication mechanism.

Traffic Filtering and Firewalls

This part describes how to configure your networking devices to filter traffic or to function as a firewall.

- Cisco implements traffic filters with access control lists (also called access lists). Access lists determine what traffic is blocked and what traffic is forwarded at router interfaces. Cisco provides both basic and advanced access list capabilities.

  — Basic access lists

  An overview of basic access lists is in the chapter "Access Control Lists: Overview and Guidelines." This chapter describes tips, cautions, considerations, recommendations, and general guidelines for configuring access lists for the various network protocols. You should configure basic access lists for all network protocols that will be routed through your networking device, such as IP, IPX, AppleTalk, and so forth.

  — Advanced access lists

  The advanced access list capabilities and configuration are described in the remaining chapters in the "Traffic Filtering and Firewalls" part of this document. The advanced access lists provide sophisticated and dynamic traffic filtering capabilities for stronger, more flexible network security.

- Cisco Secure Integrated Software (IS) provides an extensive set of security features, allowing you to configure a simple or elaborate firewall, according to your particular requirements. The following features are key components of Cisco Secure IS:

  — Context-based Access Control (CBAC)

  CBAC intelligently filtersTCP and UDP packets based on application-layer protocol session information. You can configure CBAC to permit specified TCP and UDP traffic through a firewall only when the connection is initiated from within the network you want to protect. CBAC can inspect traffic for sessions that originate from either side of the firewall, and CBAC can be used for intranet, extranet, and Internet perimeters of your network.

  — Cisco Secure Integrated Software Intrusion Detection System (IDS)

  The Cisco Secure Integrated Software IDS supports intrusion detection technology for mid-range and high-end router platforms with firewall support. It identifies 59 of the most common attacks using "signatures" to detect patterns of misuse in network traffic. The Cisco Secure IS IDS acts as an in-line intrusion detection sensor, watching packets and sessions as they flow through the router, scanning each to match any of the IDS signatures. When it detects suspicious activity, it responds before network security can be compromised and logs the event through Cisco IOS syslog.

  Cisco Secure Integrated Software IDS is compatible with the Cisco Secure Intrusion Detection System (formally known as NetRanger)—an enterprise-scale, real-time, intrusion detection system designed to detect, report, and terminate unauthorized activity throughout a network.

- Authentication Proxy

  The Cisco Secure Integrated Software authentication proxy feature allows network administrators to apply specific security policies on a per-user basis. Previously, user identity and related authorized access was associated with a user's IP address, or a single security policy had to be applied to an entire user group or sub network. Now, users can be identified and authorized on the basis of their per-user policy, and access privileges tailored on an individual basis are possible, as opposed to general policy applied across multiple users.

- Port to Application Mapping (PAM)

  Port to Application Mapping (PAM) is a feature of Cisco Secure Integrated Software. PAM allows you to customize TCP or UDP port numbers for network services or applications. PAM uses this information to support network environments that run services using ports that are different from the registered or well-known ports associated with an application. For example, the information in the PAM table enables Context-based Access Control (CBAC) supported services to run on non-standard ports.

Firewalls are discussed in the "Cisco Secure Integrated Software Firewall Overview" and "Configuring Context-Based Access Control" chapters.

IP Security and Encryption

This part describes how to configure IP security and encryption in the following chapters:

- Configuring Cisco Encryption Technology

  This chapter describes how to configure Cisco Encryption Technology (CET). CET provides network data encryption that is used to prevent routed traffic from being examined or tampered with while it travels across a network. This feature allows IP packets to be encrypted at a Cisco router, routed across a network as encrypted information, and decrypted at the destination Cisco router.

- Configuring IPSec Network Security

  This chapter describes how to configure IPSec. IPSec provides security for transmission of sensitive information over unprotected networks such as the Internet IPSec provides data authentication and anti-replay services in addition to data confidentiality services.

- Configuring Certification Authority Interoperability

  This chapter describes how to configure Certification Authority (CA) Interoperability. CA Interoperability permits Cisco IOS devices and CAs to communicate so that your Cisco IOS device can obtain and use digital certificates from the CA.

- Configuring Internet Key Exchange Security Protocol

  This chapter describes how to configure Internet Key Exchange (IKE). IKE is a key management protocol standard that is used in conjunction with the IPSec standard. IPSec can be configured without IKE, but IKE enhances IPSec by providing additional features, flexibility, and ease of configuration for the IPSec standard.

Other Security Features

This part describes four important security features in the following chapters:

- Configuring Passwords and Privileges

This chapter describes how to configure static passwords stored on your networking device. These passwords are used to control access to the device's command line prompt to view or change the device configuration.

This chapter also describes how to assign privilege levels to the passwords. You can configure up to 16 different privilege levels, and assign each level to a password. For each privilege level you define a subset of Cisco IOS commands that can be executed. You can use these different levels to allow some users the ability to execute all Cisco IOS commands, and to restrict other users to a defined subset of commands.

This chapter also describes how to recover lost passwords.

- Neighbor Router Authentication: Overview and Guidelines

  This chapter briefly describes the security benefits and operation of neighbor router authentication.

  When neighbor authentication is configured on a router, the router authenticates its neighbor router before accepting any route updates from that neighbor. This ensures that a router always receives reliable routing update information from a trusted source.

- Configuring IP Security Options

  This chapter describes how to configure IP Security Options (IPSO) as described in RFC 1108. IPSO is generally used to comply with the security policy of the U.S. government's Department of Defense.

- Configuring Unicast Reverse Path Forwarding

  This chapter describes the Unicast Reverse Path Forwarding (Unicast RPF) feature, which helps mitigate problems caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP packets that lack a verifiable IP source address. For example, a number of common types of denial-of-service (DoS) attacks, including Smurf and Tribe Flood Network (TFN), can take advantage of forged or rapidly changing source IP addresses to allow attackers to thwart efforts to locate or filter the attacks. For Internet service providers (ISPs) that provide public access, Unicast RPF deflects such attacks by forwarding only packets that have source addresses that are valid and consistent with the IP routing table. This action protects the network of the ISP, its customer, and the rest of the Internet.

Appendixes

This part describes the supported RADIUS attributes and TACACS+ attribute-value pairs in the following appendixes:

- RADIUS Attributes

  RADIUS attributes are used to define specific AAA elements in a user profile, which is stored on the RADIUS daemon. This appendix lists the RADIUS attributes currently supported.

- TACACS+ Attribute-Value Pairs

  TACACS+ attribute-value pairs are used to define specific AAA elements in a user profile, which is stored on the TACACS+ daemon. This appendix lists the TACACS+ attribute-value pairs currently supported.

Creating Effective Security Policies

An effective security policy works to ensure that your organization's network assets are protected from sabotage and from inappropriate access—both intentional and accidental.

All network security features should be configured in compliance with your organization's security policy. If you do not have a security policy, or if your policy is out of date, you should ensure that the policy is created or updated before you decide how to configure security on your Cisco device.

The following sections provide guidelines to help you create an effective security policy:

- The Nature of Security Policies
- Two Levels of Security Policies
- Tips for Developing an Effective Security Policy

The Nature of Security Policies

You should recognize these aspects of security policies:

- Security policies represent trade-offs.

  With all security policies, there is some trade-off between user productivity and security measures which can be restrictive and time consuming. The goal of any security design is to provide maximum security with minimum impact on user access and productivity. Some security measures, such as network data encryption, do not restrict access and productivity. On the other hand, cumbersome or unnecessarily redundant verification and authorization systems can frustrate users and even prevent access to critical network resources.

- Security policies should be determined by business needs.

  Business needs should dictate the security policy; a security policy should not determine how a business operates.

- Security policies are living documents.

  Because organizations are constantly subject to change, security policies must be systematically updated to reflect new business directions, technological changes, and resource allocations.

Two Levels of Security Policies

You can think of a security policy as having two levels: a requirements level and an implementation level.

- At the requirements level, a policy defines the degree to which your network assets must be protected against intrusion or destruction and also estimates the cost (consequences) of a security breach. For example, the policy could state that only human resources personnel should be able to access personnel records, or that only IS personnel should be able to configure the backbone routers. The policy could also address the consequences of a network outage (due to sabotage), or the consequences of sensitive information inadvertently being made public.
- At the implementation level, a policy defines guidelines to implement the requirements-level policy, using specific technology in a predefined way. For example, the implementation-level policy could require access lists to be configured so that only traffic from human resources host computers can access the server containing personnel records.

When creating a policy, define security requirements before defining security implementations so that you do not end up merely justifying particular technical solutions that might not actually be required.

Tips for Developing an Effective Security Policy

To develop an effective security policy, consider the recommendations in the following sections:

- Identifying Your Network Assets to Protect
- Determining Points of Risk
- Limiting the Scope of Access
- Identifying Assumptions
- Determining the Cost of Security Measures
- Considering Human Factors
- Keeping a Limited Number of Secrets
- Implementing Pervasive and Scalable Security
- Understanding Typical Network Functions
- Remembering Physical Security

Identifying Your Network Assets to Protect

The first step to developing a security policy is to understand and identify your organization's network assets. Network assets include the following:

- Networked hosts (such as PCs; includes the hosts' operating systems, applications, and data)
- Networking devices (such as routers)
- Network data (data that travels across the network)

You must both identify your network's assets and determine the degree to which each of these assets must be protected. For example, one subnetwork of hosts might contain extremely sensitive data that should be protected at all costs, while a different subnetwork of hosts might require only modest protection against security risks because there is less cost involved if the subnetwork is compromised.

Determining Points of Risk

You must understand how potential intruders can enter your organization's network or sabotage network operation. Special areas of consideration are network connections, dial-up access points, and misconfigured hosts. Misconfigured hosts, frequently overlooked as points of network entry, can be systems with unprotected login accounts (guest accounts), employ extensive trust in remote commands (such as rlogin and rsh), have illegal modems attached to them, and use easy-to-break passwords.

Limiting the Scope of Access

Organizations can create multiple barriers within networks, so that unlawful entry to one part of the system does not automatically grant entry to the entire infrastructure. Although maintaining a high level of security for the entire network can be prohibitively expensive (in terms of systems and equipment as well as productivity), you can often provide higher levels of security to the more sensitive areas of your network.

Identifying Assumptions

Every security system has underlying assumptions. For example, an organization might assume that its network is not tapped, that intruders are not very knowledgeable, that intruders are using standard software, or that a locked room is safe. It is important to identify, examine, and justify your assumptions: any hidden assumption is a potential security hole.

Determining the Cost of Security Measures

In general, providing security comes at a cost. This cost can be measured in terms of increased connection times or inconveniences to legitimate users accessing the assets, or in terms of increased network management requirements, and sometimes in terms of actual dollars spent on equipment or software upgrades.

Some security measures inevitably inconvenience some sophisticated users. Security can delay work, create expensive administrative and educational overhead, use significant computing resources, and require dedicated hardware.

When you decide which security measures to implement, you must understand their costs and weigh these against potential benefits. If the security costs are out of proportion to the actual dangers, it is a disservice to the organization to implement them.

Considering Human Factors

If security measures interfere with essential uses of the system, users resist these measures and sometimes even circumvent them. Many security procedures fail because their designers do not take this fact into account. For example, because automatically generated "nonsense" passwords can be difficult to remember, users often write them on the undersides of keyboards. A "secure" door that leads to a system's only tape drive is sometimes propped open. For convenience, unauthorized modems are often connected to a network to avoid cumbersome dial-in security procedures. To ensure compliance with your security measures, users must be able to get their work done as well as understand and accept the need for security.

Any user can compromise system security to some degree. For example, an intruder can often learn passwords by simply calling legitimate users on the telephone claiming to be a system administrator and asking for them. If users understand security issues and understand the reasons for them, they are far less likely to compromise security in this way.

Defining such human factors and any corresponding policies needs to be included as a formal part of your complete security policy.

At a minimum, users must be taught never to release passwords or other secrets over unsecured telephone lines (especially through cordless or cellular telephones) or electronic mail. They 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 in which employees are not allowed access to the network until they have completed a formal training program.

Keeping a Limited Number of Secrets

Most security is based on secrets; for example, passwords and encryption keys are secrets. But the more secrets there are, the harder it is to keep all of them. It is prudent, therefore, to design a security policy that relies on a limited number of secrets. Ultimately, the most important secret an organization has is the information that can help someone circumvent its security.

Implementing Pervasive and Scalable Security

Use a systematic approach to security that includes multiple, overlapping security methods.

Almost any change that is made to a system can affect security. This is especially true when new services are created. System administrators, programmers, and users need to consider the security implications of every change they make. Understanding the security implications of a change takes practice; it requires lateral thinking and a willingness to explore every way that a service could potentially be manipulated. The goal of any security policy is to create an environment that is not susceptible to every minor change.

Understanding Typical Network Functions

Understand how your network system normally functions, know what is expected and unexpected behavior, and be familiar with how devices are usually used. This kind of awareness helps the organization detect security problems. Noticing unusual events can help catch intruders before they can damage the system. Software auditing tools can help detect, log, and track unusual events. In addition, an organization should know exactly what software it relies on to provide auditing trails, and a security system should not operate on the assumption that all software is bug free.

Remembering Physical Security

The physical security of your network devices and hosts cannot be neglected. For example, many facilities implement physical security by using security guards, closed circuit television, card-key entry systems, or other means to control physical access to network devices and hosts. Physical access to a computer or router usually gives a sophisticated user complete control over that device. Physical access to a network link usually allows a person to tap into that link, jam it, or inject traffic into it. Software security measures can often be circumvented when access to the hardware is not controlled.

Identifying Security Risks and Cisco IOS Solutions

Cisco IOS software provides a comprehensive set of security features to guard against specific security risks. This section describes a few common security risks that might be present in your network, and describes how to use Cisco IOS software to protect against each of these risks:

- Preventing Unauthorized Access into Networking Devices
- Preventing Unauthorized Access into Networks
- Preventing Network Data Interception
- Preventing Fraudulent Route Updates

Preventing Unauthorized Access into Networking Devices

If someone were to gain console or terminal access into a networking device, such as a router, switch, or network access server, that person could do significant damage to your network—perhaps by reconfiguring the device, or even by simply viewing the device's configuration information.

Typically, you want administrators to have access to your networking device; you do not want other users on your local-area network or those dialing in to the network to have access to the router.

Users can access Cisco networking devices by dialing in from outside the network through an asynchronous port, connecting from outside the network through a serial port, or connecting via a terminal or workstation from within the local network.

To prevent unauthorized access into a networking device, you should configure one or more of the following security features:

- At a minimum, you should configure passwords and privileges at each networking device for all device lines and ports, as described in the chapter "Configuring Passwords and Privileges." These passwords are stored on the networking device. When users attempt to access the device through a particular line or port, they must enter the password applied to the line or port before they can access the device.
- For an additional layer of security, you can also configure username/password pairs, stored in a database on the networking device, as described in the chapter "Configuring Passwords and Privileges." These pairs are assigned to lines or interfaces and authenticate each user before that user can access the device. If you have defined privilege levels, you can also assign a specific privilege level (with associated rights and privileges) to each username/password pair.
- If you want to use username/password pairs, but you want to store them centrally instead of locally on each individual networking device, you can store them in a database on a security server. Multiple networking devices can then use the same database to obtain user authentication (and, if necessary, authorization) information. Cisco supports a variety of security server protocols, such as RADIUS, TACACS+, and Kerberos. If you decide to use the database on a security server to store login username/password pairs, you must configure your router or access server to support the applicable protocol; in addition, because most supported security protocols must be administered through the AAA security services, you will probably need to enable AAA. For more information about security protocols and AAA, refer to the chapters in the "Authentication, Authorization, and Accounting (AAA)" part of this document.

Note Cisco recommends that, whenever possible, AAA be used to implement authentication.

- If you want to authorize individual users for specific rights and privileges, you can implement AAA's authorization feature, using a security protocol such as TACACS+ or RADIUS. For more information about security protocol features and AAA, refer to the chapters in the "Authentication, Authorization, and Accounting (AAA)" part of this document.
- If you want to have a backup authentication method, you must configure AAA. AAA allows you to specify the primary method for authenticating users (for example, a username/password database stored on a TACACS+ server) and then specify backup methods (for example, a locally stored username/password database.) The backup method is used if the primary method's database cannot be accessed by the networking device. To configure AAA, refer to the chapters in the "Authentication, Authorization, and Accounting (AAA)" part of this document. You can configure up to four sequential backup methods.

Note If you do not have backup methods configured, you will be denied access to the device if the username/password database cannot be accessed for any reason.

- If you want to keep an audit trail of user access, configure AAA accounting as described in the chapter "Configuring Accounting."

Preventing Unauthorized Access into Networks

If someone were to gain unauthorized access to your organization's internal network, that person could cause damage in many ways, perhaps by accessing sensitive files from a host, by planting a virus, or by hindering network performance by flooding your network with illegitimate packets.

This risk can also apply to a person within your network attempting to access another internal network such as a Research and Development subnetwork with sensitive and critical data. That person could intentionally or inadvertently cause damage; for example, that person might access confidential files or tie up a time-critical printer.

To prevent unauthorized access through a networking device into a network, you should configure one or more of these security features:

- Traffic Filtering

  Cisco uses access lists to filter traffic at networking devices. Basic access lists allow only specified traffic through the device; other traffic is simply dropped. You can specify individual hosts or subnets that should be allowed into the network, and you can specify what type of traffic should be allowed into the network. Basic access lists generally filter traffic based on source and destination addresses, and protocol type of each packet.

  Advanced traffic filtering is also available, providing additional filtering capabilities; for example, the Lock-and-Key Security feature requires each user to be authenticated via a username/password before that user's traffic is allowed onto the network.

  All the Cisco IOS traffic filtering capabilities are described in the chapters in the "Traffic Filtering and Firewalls" part of this document.

- Authentication

  You can require users to be authenticated before they gain access into a network. When users attempt to access a service or host (such as a web site or file server) within the protected network, they must first enter certain data such as a username and password, and possibly additional information such as their date of birth or mother's maiden name. After successful authentication (depending on the method of authentication), users will be assigned specific privileges, allowing them to access specific network assets. In most cases, this type of authentication would be facilitated by using CHAP or PAP over a serial PPP connection in conjunction with a specific security protocol, such as TACACS+ or RADIUS.

  Just as in preventing unauthorized access to specific network devices, you need to decide whether or not you want the authentication database to reside locally or on a separate security server. In this case, a local security database is useful if you have very few routers providing network access. A local security database does not require a separate (and costly) security server. A remote, centralized security database is convenient when you have a large number of routers providing network access because it prevents you from having to update each router with new or changed username authentication and authorization information for potentially hundreds of thousands of dial-in users. A centralized security database also helps establish consistent remote access policies throughout a corporation.

  Cisco IOS software supports a variety of authentication methods. Although AAA is the primary (and recommended) method for access control, Cisco IOS software provides additional features for simple access control that are outside the scope of AAA. For more information, refer to the "Configuring Authentication" chapter.

Preventing Network Data Interception

When packets travel across a network, they are susceptible to being read, altered, or "hijacked." (Hijacking occurs when a hostile party intercepts a network traffic session and poses as one of the session endpoints.)

If the data is traveling across an unsecured network such as the Internet, the data is exposed to a fairly significant risk. Sensitive or confidential data could be exposed, critical data could be modified, and communications could be interrupted if data is altered.

To protect data as it travels across a network, configure network data encryption, as described in the chapter "Configuring IPSec Network Security."

IPSec provides the following network security services. These services are optional. In general, local security policy will dictate the use of one or more of the following services:

- Data Confidentiality—The IPSec sender can encrypt packets before transmitting them across a network.
- Data Integrity—The IPSec receiver can authenticate packets sent by the IPSec sender to ensure that the data has not been altered during transmission.
- Data Origin Authentication—The IPSec receiver can authenticate the source of the IPSec packets sent. This service is dependent upon the data integrity service.
- Anti-Replay—The IPSec receiver can detect and reject replayed packets.

Cisco IPSec prevents routed traffic from being examined or tampered with while it travels across a network. This feature causes IP packets to be encrypted at a Cisco router, routed across a network as encrypted information, and decrypted at the destination Cisco router. In between the two routers, the packets are in encrypted form and therefore the packets' contents cannot be read or altered. You define what traffic should be encrypted between the two routers, according to what data is more sensitive or critical.

If you want to protect traffic for protocols other than IP, you can encapsulate those other protocols into IP packets using GRE encapsulation, and then encrypt the IP packets.

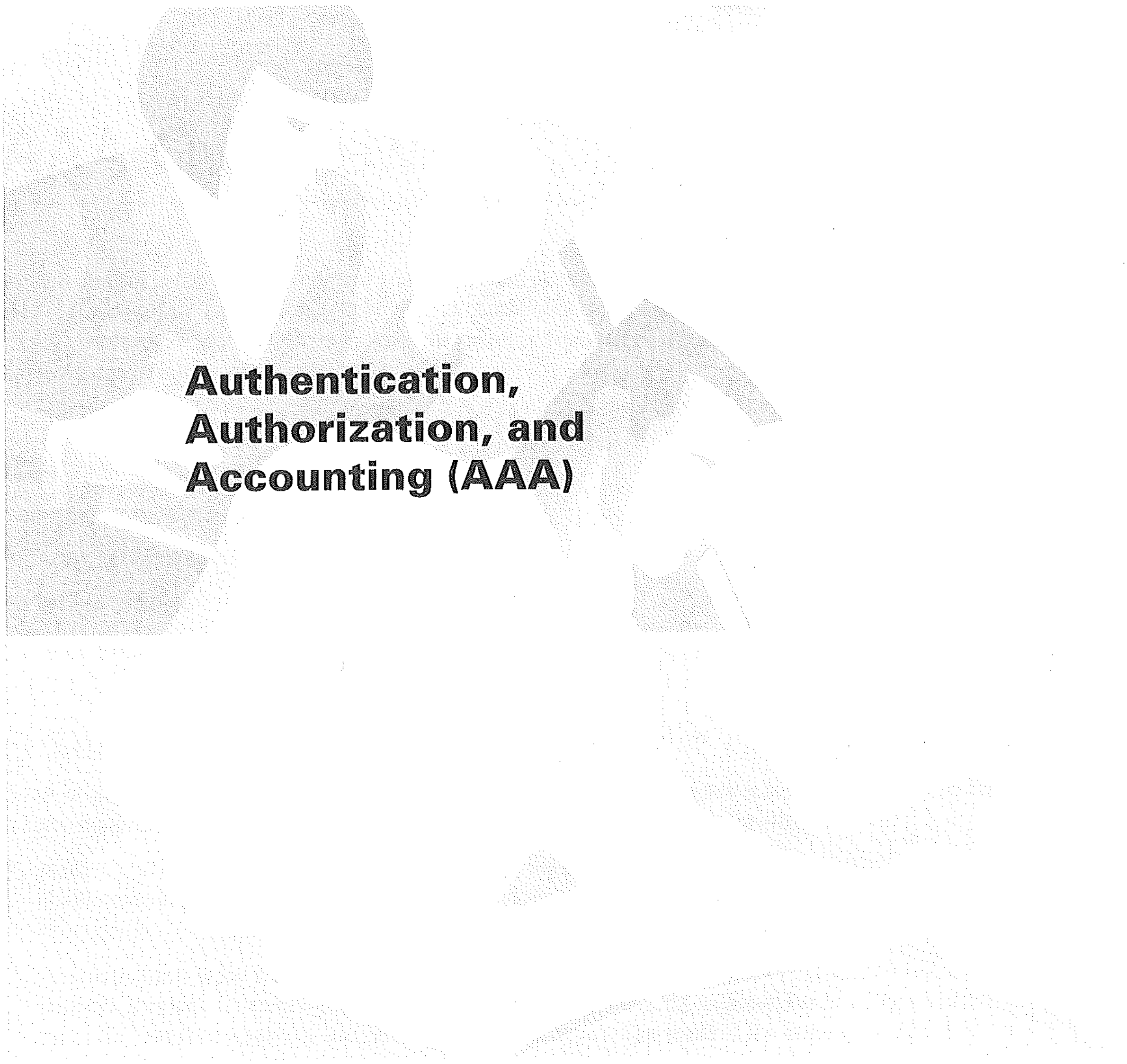
Typically, you do not use IPSec for traffic that is routed through networks that you consider secure. Consider using IPSec for traffic that is routed across unsecured networks, such as the Internet, if your organization could be damaged if the traffic is examined or tampered with by unauthorized individuals.

Preventing Fraudulent Route Updates

All routing devices determine where to route individual packets by using information stored in route tables. This route table information is created using route updates obtained from neighboring routers.

If a router receives a fraudulent update, the router could be tricked into forwarding traffic to the wrong destination. This could cause sensitive data to be exposed, or could cause network communications to be interrupted.

To ensure that route updates are received only from known, trusted neighbor routers, configure neighbor router authentication as described in the chapter "Neighbor Router Authentication: Overview and Guidelines."



Authentication, Authorization, and Accounting (AAA)

AAA Overview

Access control is the way you control who is allowed access to the network server and what services they are allowed to use once they have access. Authentication, Authorization, and Accounting (AAA) network security services provide the primary framework through which you set up access control on your router or access server.

In This Chapter

This chapter includes the following sections:

- AAA Security Services
- Where to Begin
- What to Do Next

AAA Security Services

AAA is an architectural framework for configuring a set of three independent security functions in a consistent manner. AAA provides a modular way of performing the following services:

- Authentication—Provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol you select, encryption.

  Authentication is the way a user is identified prior to being allowed access to the network and network services. You configure AAA authentication by defining a named list of authentication methods, and then applying that list to various interfaces. The method list defines the types of authentication to be performed and the sequence in which they will be performed; it must be applied to a specific interface before any of the defined authentication methods will be performed. The only exception is the default method list (which, by coincidence, is named "default"). The default method list is automatically applied to all interfaces if no other method list is defined. A defined method list overrides the default method list.

  All authentication methods, except for local, line password, and enable authentication, must be defined through AAA. For information about configuring all authentication methods, including those implemented outside of the AAA security services, refer to the "Configuring Authentication" chapter.

- Authorization—Provides the method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, ARA, and Telnet.

AAA authorization works by assembling a set of attributes that describe what the user is authorized to perform. These attributes are compared to the information contained in a database for a given user and the result is returned to AAA to determine the user's actual capabilities and restrictions. The database can be located locally on the access server or router or it can be hosted remotely on a RADIUS or TACACS+ security server. Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights, with the appropriate user. All authorization methods must be defined through AAA.

As with authentication, you configure AAA authorization by defining a named list of authorization methods, and then applying that list to various interfaces. For information about configuring authorization using AAA, refer to the "Configuring Authorization" chapter.

- Accounting—Provides the method for collecting and sending security server information used for billing, auditing, and reporting, such as user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes.

  Accounting enables you to track the services users are accessing as well as the amount of network resources they are consuming. When AAA accounting is activated, the network access server reports user activity to the TACACS+ or RADIUS security server (depending on which security method you have implemented) in the form of accounting records. Each accounting record is comprised of accounting AV pairs and is stored on the access control server. This data can then be analyzed for network management, client billing, and/or auditing. All accounting methods must be defined through AAA. As with authentication and authorization, you configure AAA accounting by defining a named list of accounting methods, and then applying that list to various interfaces. For information about configuring accounting using AAA, refer to the "Configuring Accounting" chapter.

In many circumstances, AAA uses protocols such as RADIUS, TACACS+, and Kerberos to administer its security functions. If your router or access server is acting as a network access server, AAA is the means through which you establish communication between your network access server and your RADIUS, TACACS+, or Kerberos security server.

Although AAA is the primary (and recommended) method for access control, Cisco IOS software provides additional features for simple access control that are outside the scope of AAA, such as local username authentication, line password authentication, and enable password authentication. However, these features do not provide the same degree of access control that is possible by using AAA.

This section includes the following sections:

- Benefits of Using AAA
- AAA Philosophy
- Method Lists

Benefits of Using AAA

AAA provides the following benefits:

- Increased flexibility and control
- Scalability
- Standardized authentication methods, such as RADIUS, TACACS+, and Kerberos
- Multiple backup systems

Note The deprecated protocols, TACACS and extended TACACS, are not compatible with AAA; if you select these security protocols, you will not be able to take advantage of the AAA security services.

AAA Philosophy

AAA is designed to enable you to dynamically configure the type of authentication and authorization you want on a per-line (per-user) or per-service (for example, IP, IPX, or VPDN) basis. You define the type of authentication and authorization you want by creating method lists, then applying those method lists to specific services or interfaces.

For information about applications that use AAA, such as per-user configuration and virtual profiles, refer to the "Configuring Per-User Configuration" and "Configuring Virtual Profiles" chapters in the *Cisco IOS Dial Services Configuration Guide: Network Services.*

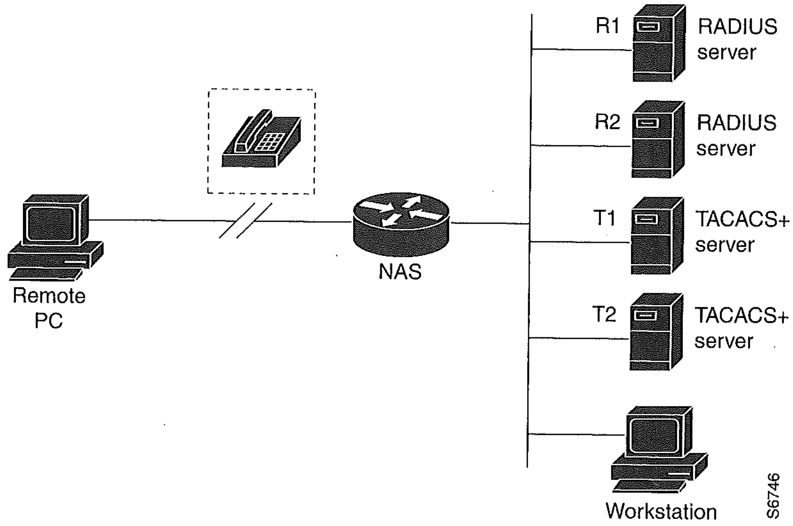
Method Lists

A method list is simply a list defining the authentication methods to be used, in sequence, to authenticate a user. Method lists enable you to designate one or more security protocols to be used for authentication, thus ensuring a backup system for authentication in case the initial method fails. Cisco IOS software uses the first method listed to authenticate users; if that method does not respond, the Cisco IOS software selects the next authentication method listed in the method list. This process continues until there is successful communication with a listed authentication method or the authentication method list is exhausted, in which case authentication fails.

Note The Cisco IOS software attempts authentication with the next listed authentication method only when there is no response from the previous method. If authentication fails at any point in this cycle—meaning that the security server or local username database responds by denying the user access—the authentication process stops and no other authentication methods are attempted.

Figure 2 shows a typical AAA network configuration that includes four security servers: R1 and R2 are RADIUS servers, and T1 and T2 are TACACS+ servers.

Figure 2 Typical AAA Network Configuration



Suppose the system administrator has defined a method list where R1 will be contacted first for authentication information, then R2, T1, T2, and then finally the local username database on the access server itself. When a remote user attempts to dial in to the network, the network access server first queries R1 for authentication information. If R1 authenticates the user, it issues a PASS response to the network access server and the user is allowed to access the network. If R1 returns a FAIL response, the user is denied access and the session is terminated. If R1 does not respond, then the network access server processes that as an ERROR and queries R2 for authentication information. This pattern would continue through the remaining designated methods until the user is either authenticated or rejected, or until the session is terminated. If all of the authentication methods return errors, which the network access server would process as a failure, the session would be terminated.

Note A FAIL response is significantly different from an ERROR. A FAIL means that the user has not met the criteria contained in the applicable authentication database to be successfully authenticated. Authentication ends with a FAIL response. An ERROR means that the security server has not responded to an authentication query. Because of this, no authentication has been attempted. Only when an ERROR is detected will AAA select the next authentication method defined in the authentication method list.

Where to Begin

You must first decide what kind of security solution you want to implement. You need to assess the security risks in your particular network and decide on the appropriate means to prevent unauthorized entry and attack. For more information about assessing your security risks and possible security solutions, refer to the "Security Overview" chapter. We recommend that you use AAA, no matter how minor your security needs might be.

This section includes the following sections:

- Overview of the AAA Configuration Process
- Enabling AAA
- Disabling AAA

Overview of the AAA Configuration Process

Configuring AAA is relatively simple after you understand the basic process involved. To configure security on a Cisco router or access server using AAA, follow this process:

1. Enable AAA by using the **aaa new-model** global configuration command.
2. If you decide to use a separate security server, configure security protocol parameters, such as RADIUS, TACACS+, or Kerberos.
3. Define the method lists for authentication by using an AAA authentication command.
4. Apply the method lists to a particular interface or line, if required.
5. (Optional) Configure authorization using the **aaa authorization** command.
6. (Optional) Configure accounting using the **aaa accounting** command.

For a complete description of the commands used in this chapter, refer to the "Authentication Commands" chapter of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Enabling AAA

Before you can use any of the services AAA network security services provide, you need to enable AAA. To enable AAA, use the following command in global configuration mode:

Command	Purpose
aaa new-model	Enables AAA.

Note When you enable AAA, you can no longer access the commands to configure the older deprecated protocols, TACACS or extended TACACS. If you decided to use TACACS or extended TACACS in your security solution, do not enable AAA.

Disabling AAA

You can disable AAA functionality with a single command if, for some reason, you decide that your security needs cannot be met by AAA but can be met by using TACACS, extended TACACS, or a line security method that can be implemented without AAA. To disable AAA, use the following command in global configuration mode:

Command	Purpose
no aaa new-model	Disables AAA.

What to Do Next

Once you have enabled AAA, you are ready to configure the other elements relating to your selected security solution. Table 3 describes AAA configuration tasks and where to find more information.

Table 3 AAA Access Control Security Solutions Methods

Task	Chapter in the *Cisco IOS Security Configuration Guide*
Configuring local login authentication	Configuring Authentication
Controlling login using security server authentication	Configuring Authentication
Defining method lists for authentication	Configuring Authentication
Applying method lists to a particular interface or line	Configuring Authentication
Configuring RADIUS security protocol parameters	Configuring RADIUS
Configuring TACACS+ security protocol parameters	Configuring TACACS+
Configuring Kerberos security protocol parameters	Configuring Kerberos
Enabling TACACS+ authorization	Configuring Authorization
Enabling RADIUS authorization	Configuring Authorization
Viewing supported IETF RADIUS attributes	RADIUS Attributes
Viewing supported vendor-specific RADIUS attributes	RADIUS Attributes
Viewing supported TACACS+ AV pairs	TACACS+ AV Pairs
Enabling accounting	Configuring Accounting

If you have elected not to use the AAA security services, see the "Configuring Authentication" chapter for the non-AAA configuration task, configuring login authentication.

Configuring Authentication

Authentication identifies users before they are allowed access to the network and network services. Basically, the Cisco IOS software implementation of authentication is divided into two main categories:

- AAA Authentication Methods Configuration Task List
- Non-AAA Authentication Methods

Authentication, for the most part, is implemented through the AAA security services. We recommend that, whenever possible, AAA be used to implement authentication.

This chapter describes both AAA and non-AAA authentication methods. For configuration examples, refer to the "Authentication Examples" section at the end of this chapter. For a complete description of the AAA commands used in this chapter, refer to the "Authentication, Authorization, and Accounting (AAA)" section of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

In This Chapter

This chapter includes the following sections:

- AAA Authentication Method Lists
- AAA Authentication Methods Configuration Task List
- Non-AAA Authentication Methods
- Authentication Examples

AAA Authentication Method Lists

To configure AAA authentication, first define a named list of authentication methods, and then apply that list to various interfaces. The method list defines the types of authentication to be performed and the sequence in which they will be performed; it must be applied to a specific interface before any of the defined authentication methods will be performed. The only exception is the default method list (which, by coincidence, is named "default"). The default method list is automatically applied to all interfaces except those that have a named method list explicitly defined. A defined method list overrides the default method list.

A method list is simply a list describing the authentication methods to be queried, in sequence, to authenticate a user. Method lists enable you to designate one or more security protocols to be used for authentication, thus ensuring a backup system for authentication in case the initial method fails.

Cisco IOS software uses the first method listed to authenticate users; if that method fails to respond, the Cisco IOS software selects the next authentication method listed in the method list. This process continues until there is successful communication with a listed authentication method, or all methods defined are exhausted.

It is important to note that the Cisco IOS software attempts authentication with the next listed authentication method only when there is no response from the previous method. If authentication fails at any point in this cycle—meaning that the security server or local username database responds by denying the user access—the authentication process stops and no other authentication methods are attempted.

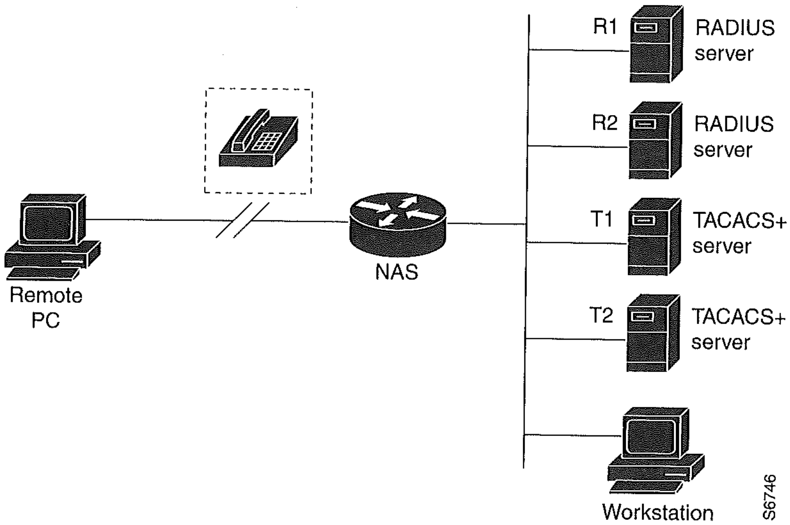
This section includes the following sections:

- Method Lists and Server Groups
- Method List Examples
- AAA Authentication General Configuration Procedure

Method Lists and Server Groups

A server group is a way to group existing RADIUS or TACACS+ server hosts for use in method lists. Figure 3 shows a typical AAA network configuration that includes four security servers: R1 and R2 are RADIUS servers and T1 and T2 are TACACS+ servers. R1 and R2 comprise the group of RADIUS server. T1 and T2 comprise the group of TACACS+ servers.

Figure 3 **Typical AAA Network Configuration**



Using server groups, you can specify a subset of the configured server hosts and use them for a particular service. For example, server groups allow you to define R1 and R2 as separate server groups, and T1 and T2 as separate server groups. This way you can specify R1 and T1 in the method list for authentication login, while specifying R2 and T2 in the method list for PPP authentication.

Server groups also can include multiple host entries for the same server, as long as each entry has a unique identifier. The combination of an IP address and a UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different

UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, authentication—the second host entry configured acts as fail-over backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server will try the second host entry configured on the same device for accounting services. (The RADIUS host entries will be tried in the order they are configured.)

For more information about configuring server groups and about configuring server groups based on DNIS numbers, refer to the "Configuring RADIUS" or "Configuring TACACS+" chapter.

Method List Examples

Figure 3 shows a typical AAA network configuration that includes four security servers: R1 and R2 are RADIUS servers, and T1 and T2 are TACACS+ servers. R1 and R2 comprise the group of RADIUS servers. T1 and T2 comprise the group of TACACS+ servers. Suppose the system administrator has decided on a security solution where all interfaces will use the same authentication methods to authenticate PPP connections: in the RADIUS group, R1 is contacted first for authentication information, then if there is no response, R2 is contacted. If R2 does not respond, T1 in the TACACS+ group is contacted; if T1 does not respond, T2 is contacted. If all designated servers fail to respond, authentication falls over to the local username database on the access server itself. To implement this, the system administrator would create a default method list by entering the following command:

```
aaa authentication ppp default group radius group tacacs+ local
```

In this example, "default" is the name of the method list. The protocols included in this method list are listed after the name, in the order they are to be queried. The default list is automatically applied to all interfaces.

When a remote user attempts to dial in to the network, the network access server first queries R1 for authentication information. If R1 authenticates the user, it issues a PASS response to the network access server and the user is allowed to access the network. If R1 returns a FAIL response, the user is denied access and the session is terminated. If R1 does not respond, then the network access server processes that as an ERROR and queries R2 for authentication information. This pattern would continue through the remaining designated methods until the user is either authenticated or rejected, or until the session is terminated.

It is important to remember that a FAIL response is significantly different from an ERROR. A FAIL means that the user has not met the criteria contained in the applicable authentication database to be successfully authenticated. Authentication ends with a FAIL response. An ERROR means that the security server has not responded to an authentication query. Because of this, no authentication has been attempted. Only when an ERROR is detected will AAA select the next authentication method defined in the authentication method list.

Suppose the system administrator wanted to apply a method list only to a particular interface or set of interfaces. In this case, the system administrator would create a named method list and then apply this named list to the applicable interfaces. The following example shows how the system administrator would implement an authentication method that will be applied only to interface 3:

```
aaa authentication ppp default group radius group tacacs+ local
aaa authentication ppp apple group radius group tacacs+ local none
 interface async 3
 ppp authentication chap apple
```

In this example, "apple" is the name of the method list, and the protocols included in this method list are listed after the name in the order in which they are to be performed. After the method list has been created, it is applied to the appropriate interface. Note that the method list name (apple) in both the AAA and PPP authentication commands must match.

In the following example, the system administrator uses server groups to specify that only R2 and T2 are valid servers for PPP authentication. To do this, the administrator must define specific server groups whose members are R2 (172.16.2.7) and T2 (172.16.2.77), respectively. In this example, the RADIUS server group "rad2only" is defined as follows using the **aaa group server** command:

```
aaa group server radius rad2only
 server 172.16.2.7
```

The TACACS server group "tac2only" is defined as follows using the **aaa group server** command:

```
aaa group server tacacs+ tac2only
 server 172.16.2.77
```

The administrator then applies PPP authentication using the server groups. In this example, the default methods list for PPP authentication follows this order: **group rad2only**, **group tac2only**, and **local**:

```
aaa authentication ppp default group rad2only group tac2only local
```

AAA Authentication General Configuration Procedure

To configure AAA authentication, you need to perform the following tasks:

1. Enable AAA by using the **aaa new-model** global configuration command. For more information about configuring AAA, refer to the "AAA Overview" chapter.
2. Configure security protocol parameters, such as RADIUS, TACACS+, or Kerberos, if you are using a security server. For more information about RADIUS, refer to the "Configuring RADIUS" chapter. For more information about TACACS+, refer to the "Configuring TACACS+" chapter. For more information about Kerberos, refer to the "Configuring Kerberos" chapter.
3. Define the method lists for authentication by using an AAA authentication command.
4. Apply the method lists to a particular interface or line, if required.

AAA Authentication Methods Configuration Task List

This section discusses the following AAA authentication methods:

- Configuring Login Authentication Using AAA
- Configuring PPP Authentication Using AAA
- Configuring AAA Scalability for PPP Requests
- Configuring ARA Authentication Using AAA
- Configuring NASI Authentication Using AAA
- Specifying the Amount of Time for Login Input
- Enabling Password Protection at the Privileged Level
- Changing the Text Displayed at the Password Prompt
- Configuring Message Banners for AAA Authentication

- Enabling Double Authentication
- Enabling Automated Double Authentication

Note AAA features are not available for use until you enable AAA globally by issuing the **aaa new-model** command. For more information about enabling AAA, refer to the "AAA Overview" chapter.

For authentication configuration examples using the commands in this chapter, refer to the "Authentication Examples" section at the end of the this chapter.

Configuring Login Authentication Using AAA

The AAA security services facilitate a variety of login authentication methods. Use the **aaa authentication login** command to enable AAA authentication no matter which of the supported login authentication methods you decide to use. With the **aaa authentication login** command, you create one or more lists of authentication methods that are tried at login. These lists are applied using the **login authentication** line configuration command.

To configure login authentication by using AAA, use the following commands beginning in global configuration mode:

Step	Command	Purpose
1	`aaa new-model`	Enables AAA globally.
2	`aaa authentication login {default | list-name} method1 [method2...]`	Creates a local authentication list.
3	`line [aux | console | tty | vty] line-number [ending-line-number]`	Enters line configuration mode for the lines to which you want to apply the authentication list.
4	`login authentication {default | list-name}`	Applies the authentication list to a line or set of lines.

The *list-name* is a character string used to name the list you are creating. The method argument refers to the actual method the authentication algorithm tries. The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

For example, to specify that authentication should succeed even if (in this example) the TACACS+ server returns an error, enter the following command:

```
aaa authentication login default group tacacs+ none
```

Note Because the **none** keyword enables *any* user logging in to successfully authenticate, it should be used only as a backup method of authentication.

To create a default list that is used when a named list is *not* specified in the **login authentication** command, use the **default** keyword followed by the methods that are to be used in default situations. The default method list is automatically applied to all interfaces.

For example, to specify RADIUS as the default method for user authentication during login, enter the following command:

```
aaa authentication login default group radius
```

Table 4 lists the supported login authentication methods.

Table 4 AAA Authentication Login Methods

Keyword	Description
enable	Uses the enable password for authentication.
krb5	Uses Kerberos 5 for authentication.
krb5-telnet	Uses Kerberos 5 Telnet authentication protocol when using Telnet to connect to the router. If selected, this keyword must be listed as the first method in the method list.
line	Uses the line password for authentication.
local	Uses the local username database for authentication.
local-case	Uses case-sensitive local username authentication.
none	Uses no authentication.
group radius	Uses the list of all RADIUS servers for authentication.
group tacacs+	Uses the list of all TACACS+ servers for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

Note The **login** command only changes username and privilege level but it does not execute a shell; therefore autocommands will not be executed. To execute autocommands under this circumstance, you need to establish a Telnet session back into the router (loop-back). Make sure that the router has been configured for secure Telnet sessions if you choose to implement autocommands this way.

This section includes the following sections:

- Login Authentication Using Enable Password
- Login Authentication Using Kerberos
- Login Authentication Using Line Password
- Login Authentication Using Local Password
- Login Authentication Using Group RADIUS
- Login Authentication Using Group TACACS+
- Login Authentication Using Group group-name

Login Authentication Using Enable Password

Use the **aaa authentication login** command with the **enable** *method* keyword to specify the enable password as the login authentication method. For example, to specify the enable password as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default enable
```

Before you can use the enable password as the login authentication method, you need to define the enable password. For more information about defining enable passwords, refer to the "Configuring Passwords and Privileges" chapter.

Login Authentication Using Kerberos

Authentication via Kerberos is different from most other authentication methods: the user's password is never sent to the remote access server. Remote users logging in to the network are prompted for a username. If the key distribution center (KDC) has an entry for that user, it creates an encrypted ticket granting ticket (TGT) with the password for that user and sends it back to the router. The user is then prompted for a password, and the router attempts to decrypt the TGT with that password. If it succeeds, the user is authenticated and the TGT is stored in the user's credential cache on the router.

A user does not need to run the KINIT program to get a TGT to authenticate to the router. This is because KINIT has been integrated into the login procedure in the Cisco IOS implementation of Kerberos.

Use the **aaa authentication login** command with the **krb5** *method* keyword to specify Kerberos as the login authentication method. For example, to specify Kerberos as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default krb5
```

Before you can use Kerberos as the login authentication method, you need to enable communication with the Kerberos security server. For more information about establishing communication with a Kerberos server, refer to the "Configuring Kerberos" chapter.

Login Authentication Using Line Password

Use the **aaa authentication login** command with the **line** *method* keyword to specify the line password as the login authentication method. For example, to specify the line password as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default line
```

Before you can use a line password as the login authentication method, you need to define a line password. For more information about defining line passwords, refer to the "Configuring Line Password Protection" section in this chapter.

Login Authentication Using Local Password

Use the **aaa authentication login** command with the **local** *method* keyword to specify that the Cisco router or access server will use the local username database for authentication. For example, to specify the local username database as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default local
```

For information about adding users into the local username database, refer to the "Establishing Username Authentication" section in this chapter.

Login Authentication Using Group RADIUS

Use the **aaa authentication login** command with the **group radius** *method* to specify RADIUS as the login authentication method. For example, to specify RADIUS as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default group radius
```

Before you can use RADIUS as the login authentication method, you need to enable communication with the RADIUS security server. For more information about establishing communication with a RADIUS server, refer to the "Configuring RADIUS" chapter.

Login Authentication Using Group TACACS+

Use the **aaa authentication login** command with the **group tacacs+** *method* to specify TACACS+ as the login authentication method. For example, to specify TACACS+ as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default group tacacs+
```

Before you can use TACACS+ as the login authentication method, you need to enable communication with the TACACS+ security server. For more information about establishing communication with a TACACS+ server, refer to the "Configuring TACACS+" chapter.

Login Authentication Using Group *group-name*

Use the **aaa authentication login** command with the **group** *group-name* method to specify a subset of RADIUS or TACACS+ servers to use as the login authentication method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group loginrad**:

```
aaa group server radius loginrad
 server 172.16.2.3
 server 172.16.2 17
 server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *loginrad*.

To specify **group loginrad** as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default group loginrad
```

Before you can use a group name as the login authentication method, you need to enable communication with the RADIUS or TACACS+ security server. For more information about establishing communication with a RADIUS server, refer to the "Configuring RADIUS" chapter. For more information about establishing communication with a TACACS+ server, refer to the "Configuring TACACS+" chapter.

Configuring PPP Authentication Using AAA

Many users access network access servers through dialup via async or ISDN. Dialup via async or ISDN bypasses the CLI completely; instead, a network protocol (such as PPP or ARA) starts as soon as the connection is established.

The AAA security services facilitate a variety of authentication methods for use on serial interfaces running PPP. Use the **aaa authentication ppp** command to enable AAA authentication no matter which of the supported PPP authentication methods you decide to use.

To configure AAA authentication methods for serial lines using PPP, use the following commands in global configuration mode:

Step	Command	Purpose
1	`aaa new-model`	Enables AAA globally.
2	`aaa authentication ppp {default	list-name} method1 [method2...]`	Creates a local authentication list.
3	`interface interface-type interface-number`	Enters interface configuration mode for the interface to which you want to apply the authentication list.
4	`ppp authentication {protocol1 [protocol2...]} [if-needed] {default	list-name} [callin] [one-time]`	Applies the authentication list to a line or set of lines. In this command, *protocol1* and *protocol2* represent the following protocols: CHAP, MS-CHAP, and PAP. PPP authentication is attempted first using the first authentication method, specified by *protocol1*. If *protocol1* is unable to establish authentication, the next configured protocol is used to negotiate authentication.

With the **aaa authentication ppp** command, you create one or more lists of authentication methods that are tried when a user tries to authenticate via PPP. These lists are applied using the **ppp authentication** line configuration command.

To create a default list that is used when a named list is *not* specified in the **ppp authentication** command, use the **default** keyword followed by the methods you want used in default situations.

For example, to specify the local username database as the default method for user authentication, enter the following command:

```
aaa authentication ppp default local
```

The *list-name* is any character string used to name the list you are creating. The method argument refers to the actual method the authentication algorithm tries. The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

For example, to specify that authentication should succeed even if (in this example) the TACACS+ server returns an error, enter the following command:

```
aaa authentication ppp default group tacacs+ none
```

Note Because **none** allows all users logging in to authenticate successfully, it should be used as a backup method of authentication.

Table 5 lists the supported login authentication methods.

Table 5 AAA Authentication PPP Methods

Keyword	Description
if-needed	Does not authenticate if user has already been authenticated on a TTY line.
krb5	Uses Kerberos 5 for authentication (can only be used for PAP authentication).
local	Uses the local username database for authentication.
local-case	Uses case-sensitive local username authentication.
none	Uses no authentication.
group radius	Uses the list of all RADIUS servers for authentication.
group tacacs+	Uses the list of all TACACS+ servers for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

This section includes the following sections:

- PPP Authentication Using Kerberos
- PPP Authentication Using Local Password
- PPP Authentication Using Group RADIUS
- PPP Authentication Using Group TACACS+
- PPP Authentication Using Group group-name

PPP Authentication Using Kerberos

Use the **aaa authentication ppp** command with the **krb5** *method* keyword to specify Kerberos as the authentication method for use on interfaces running PPP. For example, to specify Kerberos as the method of user authentication when no other method list has been defined, enter the following command:

```
aaa authentication ppp default krb5
```

Before you can use Kerberos as the PPP authentication method, you need to enable communication with the Kerberos security server. For more information about establishing communication with a Kerberos server, refer to the "Configuring Kerberos" chapter.

Note Kerberos login authentication works only with PPP PAP authentication.

PPP Authentication Using Local Password

Use the **aaa authentication ppp** command with the *method* keyword **local** to specify that the Cisco router or access server will use the local username database for authentication. For example, to specify the local username database as the method of authentication for use on lines running PPP when no other method list has been defined, enter the following command:

```
aaa authentication ppp default local
```

For information about adding users into the local username database, refer to the "Establishing Username Authentication" section in this chapter.

PPP Authentication Using Group RADIUS

Use the **aaa authentication ppp** command with the **group radius** *method* to specify RADIUS as the login authentication method. For example, to specify RADIUS as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication ppp default group radius
```

Before you can use RADIUS as the PPP authentication method, you need to enable communication with the RADIUS security server. For more information about establishing communication with a RADIUS server, refer to the "Configuring RADIUS" chapter.

PPP Authentication Using Group TACACS+

Use the **aaa authentication ppp** command with the **group tacacs+** *method* to specify TACACS+ as the login authentication method. For example, to specify TACACS+ as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication ppp default group tacacs+
```

Before you can use TACACS+ as the PPP authentication method, you need to enable communication with the TACACS+ security server. For more information about establishing communication with a TACACS+ server, refer to the "Configuring TACACS+" chapter.

PPP Authentication Using Group *group-name*

Use the **aaa authentication ppp** command with the **group** *group-name* method to specify a subset of RADIUS or TACACS+ servers to use as the login authentication method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group ppprad**:

```
aaa group server radius ppprad
 server 172.16.2.3
 server 172.16.2 17
 server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *ppprad*.

To specify **group ppprad** as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication ppp default group ppprad
```

Before you can use a group name as the PPP authentication method, you need to enable communication with the RADIUS or TACACS+ security server. For more information about establishing communication with a RADIUS server, refer to the "Configuring RADIUS" chapter. For more information about establishing communication with a TACACS+ server, refer to the "Configuring TACACS+" chapter.

Configuring AAA Scalability for PPP Requests

You can configure and monitor the number of background processes allocated by the PPP manager in the network access server (NAS) to deal with AAA authentication and authorization requests. In previous Cisco IOS releases, only one background process was allocated to handle all AAA requests for PPP. This meant that parallelism in AAA servers could not be fully exploited. The AAA Scalability feature enables you to configure the number of processes used to handle AAA requests for PPP, thus increasing the number of users that can be simultaneously authenticated or authorized.

To allocate a specific number of background processes to handle AAA requests for PPP, use the following command in global configuration mode:

Command	Purpose
`aaa processes` *`number`*	Allocates a specific number of background processes to handle AAA authentication and authorization requests for PPP.

The argument *number* defines the number of background processes earmarked to process AAA authentication and authorization requests for PPP and can be configured for any value from 1 to 2147483647. Because of the way the PPP manager handles requests for PPP, this argument also defines the number of new users that can be simultaneously authenticated. This argument can be increased or decreased at any time.

Note Allocating additional background processes can be expensive. You should configure the minimum number of background processes capable of handling the AAA requests for PPP.

Configuring ARA Authentication Using AAA

With the **aaa authentication arap** command, you create one or more lists of authentication methods that are tried when AppleTalk Remote Access (ARA) users attempt to log in to the router. These lists are used with the **arap authentication** line configuration command.

Use the following commands starting in global configuration mode:

Step	Command	Purpose
1	`aaa new-model`	Enables AAA globally.
2	`aaa authentication arap {default | `*`list-name`*`}` *`method1`* `[`*`method2...`*`]`	Enables authentication for ARA users.
3	`line` *`number`*	(Optional) Changes to line configuration mode.
4	`autoselect arap`	(Optional) Enables autoselection of ARA.
5	`autoselect during-login`	(Optional) Starts the ARA session automatically at user login.
6	`arap authentication` *`list-name`*	(Optional—not needed if **default** is used in the **aaa authentication arap** command) Enables TACACS+ authentication for ARA on a line.

The *list-name* is any character string used to name the list you are creating. The method argument refers to the actual list of methods the authentication algorithm tries, in the sequence entered.

To create a default list that is used when a named list is *not* specified in the **arap authentication** command, use the **default** keyword followed by the methods you want to be used in default situations.

The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

Note Because **none** allows all users logging in to authenticate successfully, it should be used as a backup method of authentication.

Table 6 lists the supported login authentication methods.

Table 6 AAA Authentication ARA Methods

Keyword	Description
auth-guest	Allows guest logins only if the user has already logged in to EXEC.
guest	Allows guest logins.
line	Uses the line password for authentication.
local	Uses the local username database for authentication.
local-case	Uses case-sensitive local username authentication.
group radius	Uses the list of all RADIUS servers for authentication.
group tacacs+	Uses the list of all TACACS+ servers for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

For example, to create a default AAA authentication method list used with the ARA protocol, enter the following command:

```
aaa authentication arap default if-needed none
```

To create the same authentication method list for the ARA protocol but name the list *MIS-access,* enter the following command:

```
aaa authentication arap MIS-access if-needed none
```

This section includes the following sections:

- ARA Authentication Allowing Authorized Guest Logins
- ARA Authentication Allowing Guest Logins
- ARA Authentication Using Line Password
- ARA Authentication Using Local Password
- ARA Authentication Using Group RADIUS
- ARA Authentication Using Group TACACS+
- ARA Authentication Using Group group-name

ARA Authentication Allowing Authorized Guest Logins

Use the **aaa authentication arap** command with the **auth-guest** keyword to allow guest logins only if the user has already successfully logged in to the EXEC. This method must be the first listed in the ARA authentication method list but it can be followed by other methods if it does not succeed. For example, to allow all authorized guest logins—meaning logins by users who have already successfully logged in to the EXEC—as the default method of authentication, using RADIUS only if that method fails, enter the following command:

```
aaa authentication arap default auth-guest group radius
```

For more information about ARA authorized guest logins, refer to the "Configuring AppleTalk" chapter in the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*.

Note By default, guest logins through ARA are disabled when you initialize AAA. To allow guest logins, you must use the **aaa authentication arap** command with either the **guest** or the **auth-guest** keyword.

ARA Authentication Allowing Guest Logins

Use the **aaa authentication arap** command with the **guest** keyword to allow guest logins. This method must be the first listed in the ARA authentication method list but it can be followed by other methods if it does not succeed. For example, to allow all guest logins as the default method of authentication, using RADIUS only if that method fails, enter the following command:

```
aaa authentication arap default guest group radius
```

For more information about ARA guest logins, refer to the "Configuring AppleTalk" chapter in the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*.

ARA Authentication Using Line Password

Use the **aaa authentication arap** command with the *method* keyword **line** to specify the line password as the authentication method. For example, to specify the line password as the method of ARA user authentication when no other method list has been defined, enter the following command:

```
aaa authentication arap default line
```

Before you can use a line password as the ARA authentication method, you need to define a line password. For more information about defining line passwords, refer to the "Configuring Line Password Protection" section in this chapter.

ARA Authentication Using Local Password

Use the **aaa authentication arap** command with the *method* keyword **local** to specify that the Cisco router or access server will use the local username database for authentication. For example, to specify the local username database as the method of ARA user authentication when no other method list has been defined, enter the following command:

```
aaa authentication arap default local
```

For information about adding users to the local username database, refer to the "Establishing Username Authentication" section in this chapter.

ARA Authentication Using Group RADIUS

Use the **aaa authentication arap** command with the **group radius** *method* to specify RADIUS as the ARA authentication method. For example, to specify RADIUS as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication arap default group radius
```

Before you can use RADIUS as the ARA authentication method, you need to enable communication with the RADIUS security server. For more information about establishing communication with a RADIUS server, refer to the "Configuring RADIUS" chapter.

ARA Authentication Using Group TACACS+

Use the **aaa authentication arap** command with the **group tacacs+** *method* to specify TACACS+ as the ARA authentication method. For example, to specify TACACS+ as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication arap default group tacacs+
```

Before you can use TACACS+ as the ARA authentication method, you need to enable communication with the TACACS+ security server. For more information about establishing communication with a TACACS+ server, refer to the "Configuring TACACS+" chapter.

ARA Authentication Using Group *group-name*

Use the **aaa authentication arap** command with the **group** *group-name* method to specify a subset of RADIUS or TACACS+ servers to use as the ARA authentication method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group araprad**:

```
aaa group server radius araprad
   server 172.16.2.3
   server 172.16.2 17
   server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *araprad.*

To specify **group ararad** as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication arap default group araprad
```

Before you can use a group name as the ARA authentication method, you need to enable communication with the RADIUS or TACACS+ security server. For more information about establishing communication with a RADIUS server, refer to the "Configuring RADIUS" chapter. For more information about establishing communication with a TACACS+ server, refer to the "Configuring TACACS+" chapter.

Configuring NASI Authentication Using AAA

With the **aaa authentication nasi** command, you create one or more lists of authentication methods that are tried when NetWare Asynchronous Services Interface (NASI) users attempt to log in to the router. These lists are used with the **nasi authentication** line configuration command.

Use the following commands starting in global configuration mode:

Step	Command	Purpose
1	`aaa new-model`	Enables AAA globally.
2	`aaa authentication nasi {default | list-name} method1 [method2...]`	Enables authentication for NASI users.
3	`line number`	(Optional—not needed if **default** is used in the **aaa authentication nasi** command) Changes to line configuration mode.
4	`nasi authentication list-name`	(Optional—not needed if **default** is used in the **aaa authentication nasi** command) Enables authentication for NASI on a line.

The *list-name* is any character string used to name the list you are creating. The method argument refers to the actual list of methods the authentication algorithm tries, in the sequence entered.

To create a default list that is used when a named list is *not* specified in the **aaa authentication nasi** command, use the **default** keyword followed by the methods you want to be used in default situations.

The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

Note Because **none** allows all users logging in to authenticate successfully, it should be used as a backup method of authentication.

Table 7 lists the supported NASI authentication methods.

Table 7 **AAA Authentication NASI Methods**

Keyword	Description
enable	Uses the enable password for authentication.
line	Uses the line password for authentication.
local	Uses the local username database for authentication.
local-case	Uses case-sensitive local username authentication.
none	Uses no authentication.
group radius	Uses the list of all RADIUS servers for authentication.
group tacacs+	Uses the list of all TACACS+ servers for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

This section includes the following sections:

- NASI Authentication Using Enable Password
- NASI Authentication Using Line Password
- NASI Authentication Using Local Password
- NASI Authentication Using Group RADIUS

- NASI Authentication Using Group TACACS+
- NASI Authentication Using Group group-name

NASI Authentication Using Enable Password

Use the **aaa authentication nasi** command with the *method* keyword **enable** to specify the enable password as the authentication method. For example, to specify the enable password as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default enable
```

Before you can use the enable password as the authentication method, you need to define the enable password. For more information about defining enable passwords, refer to the "Configuring Passwords and Privileges" chapter.

NASI Authentication Using Line Password

Use the **aaa authentication nasi** command with the *method* keyword **line** to specify the line password as the authentication method. For example, to specify the line password as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default line
```

Before you can use a line password as the NASI authentication method, you need to define a line password. For more information about defining line passwords, refer to the "Configuring Line Password Protection" section in this chapter.

NASI Authentication Using Local Password

Use the **aaa authentication nasi** command with the *method* keyword **local** to specify that the Cisco router or access server will use the local username database for authentication information. For example, to specify the local username database as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default local
```

For information about adding users to the local username database, refer to the "Establishing Username Authentication" section in this chapter.

NASI Authentication Using Group RADIUS

Use the **aaa authentication nasi** command with the **group radius** *method* to specify RADIUS as the NASI authentication method. For example, to specify RADIUS as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default group radius
```

Before you can use RADIUS as the NASI authentication method, you need to enable communication with the RADIUS security server. For more information about establishing communication with a RADIUS server, refer to the "Configuring RADIUS" chapter.

NASI Authentication Using Group TACACS+

Use the **aaa authentication nasi** command with the **group tacacs+** *method* keyword to specify TACACS+ as the NASI authentication method. For example, to specify TACACS+ as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default group tacacs+
```

Before you can use TACACS+ as the authentication method, you need to enable communication with the TACACS+ security server. For more information about establishing communication with a TACACS+ server, refer to the "Configuring TACACS+" chapter.

NASI Authentication Using Group *group-name*

Use the **aaa authentication nasi** command with the **group** *group-name* method to specify a subset of RADIUS or TACACS+ servers to use as the NASI authentication method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group nasirad**:

```
aaa group server radius nasirad
   server 172.16.2.3
   server 172.16.2 17
   server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *nasirad*.

To specify **group nasirad** as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication nasi default group nasirad
```

Before you can use a group name as the NASI authentication method, you need to enable communication with the RADIUS or TACACS+ security server. For more information about establishing communication with a RADIUS server, refer to the "Configuring RADIUS" chapter. For more information about establishing communication with a TACACS+ server, refer to the "Configuring TACACS+" chapter.

Specifying the Amount of Time for Login Input

The **timeout login response** command allows you to specify how long the system will wait for login input (such as username and password) before timing out. The default login value is 30 seconds; with the **timeout login response** command, you can specify a timeout value from 1 to 300 seconds. Use the following command in global configuration mode to change the login timeout value from the default of 30 seconds:

Command	Purpose
`timeout login response` *seconds*	Specifies how long the system will wait for login information before timing out.

Enabling Password Protection at the Privileged Level

Use the **aaa authentication enable default** command to create a series of authentication methods that are used to determine whether a user can access the privileged EXEC command level. You can specify up to four authentication methods. The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

Use the following command in global configuration mode:

Command	Purpose
`aaa authentication enable default` *`method1`* [*`method2...`*]	Enables user ID and password checking for users requesting privileged EXEC level.

The method argument refers to the actual list of methods the authentication algorithm tries, in the sequence entered. Table 8 lists the supported enable authentication methods.

Table 8 AAA Authentication Enable Default Methods

Keyword	Description
enable	Uses the enable password for authentication.
line	Uses the line password for authentication.
none	Uses no authentication.
group radius	Uses the list of all RADIUS hosts for authentication.
group tacacs+	Uses the list of all TACACS+ hosts for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

Changing the Text Displayed at the Password Prompt

Use the **aaa authentication password-prompt** command to change the default text that the Cisco IOS software displays when prompting a user to enter a password. This command changes the password prompt for the enable password as well as for login passwords that are not supplied by remote security servers. The **no** form of this command returns the password prompt to the following default value:

```
Password:
```

The **aaa authentication password-prompt** command does not change any dialog that is supplied by a remote TACACS+ or RADIUS server.

Use the following command in global configuration mode:

Command	Purpose
`aaa authentication password-prompt` *`text-string`*	Changes the default text displayed when a user is prompted to enter a password.

Configuring Message Banners for AAA Authentication

AAA supports the use of configurable, personalized login and failed-login banners. You can configure message banners that will be displayed when a user logs in to the system to be authenticated using AAA and when authentication, for whatever reason, fails.

This section includes the following sections:

- Configuring a Login Banner
- Configuring a Failed-Login Banner

Configuring a Login Banner

To create a login banner, you need to configure a delimiting character, which notifies the system that the following text string is to be displayed as the banner, and then the text string itself. The delimiting character is repeated at the end of the text string to signify the end of the banner. The delimiting character can be any single character in the extended ASCII character set, but once defined as the delimiter, that character cannot be used in the text string making up the banner.

To configure a banner that will be displayed whenever a user logs in (replacing the default message for login), use the following commands in global configuration mode:

Step	Command	Purpose
1	`aaa new-model`	Enables AAA.
2	**aaa authentication banner** *delimiter string delimiter*	Creates a personalized login banner.

The maximum number of characters that can be displayed in the login banner is 2996 characters.

Configuring a Failed-Login Banner

To create a failed-login banner, you need to configure a delimiting character, which notifies the system that the following text string is to be displayed as the banner, and then the text string itself. The delimiting character is repeated at the end of the text string to signify the end of the failed-login banner. The delimiting character can be any single character in the extended ASCII character set, but once defined as the delimiter, that character cannot be used in the text string making up the banner.

To configure a message that will be displayed whenever a user fails login (replacing the default message for failed login), use the following commands in global configuration mode:

Step	Command	Purpose
1	`aaa new-model`	Enables AAA.
2	**aaa authentication fail-message** *delimiter string delimiter*	Creates a message to be displayed when a user fails login.

The maximum number of characters that can be displayed in the failed-login banner is 2996 characters.

Enabling Double Authentication

Double authentication provides additional authentication for Point-to-Point Protocol (PPP) sessions. Previously, PPP sessions could only be authenticated by using a single authentication method: either PAP or CHAP. Double authentication requires remote users to pass a second stage of authentication—after CHAP or PAP authentication—before gaining network access.

This second ("double") authentication requires a password that is known to the user but *not* stored on the user's remote host. Therefore, the second authentication is specific to a user, not to a host. This provides an additional level of security that will be effective even if information from the remote host is stolen. In addition, this also provides greater flexibility by allowing customized network privileges for each user.

The second stage authentication can use one-time passwords such as token card passwords, which are not supported by CHAP. If one-time passwords are used, a stolen user password is of no use to the perpetrator.

This section includes the following sections:

- How Double Authentication Works
- Configuring Double Authentication
- Accessing the User Profile After Double Authentication

How Double Authentication Works

With double authentication, there are two authentication/authorization stages. These two stages occur after a remote user dials in and a PPP session is initiated.

In the first stage, the user logs in using the remote host name; CHAP (or PAP) authenticates the remote host, and then PPP negotiates with AAA to authorize the remote host. In this process, the network access privileges associated with the remote host are assigned to the user.

Note We suggest that the network administrator restrict authorization at this first stage to allow only Telnet connections to the local host.

In the second stage, the remote user must Telnet to the network access server to be authenticated. When the remote user logs in, the user must be authenticated with AAA login authentication. The user then must enter the **access-profile** command to be reauthorized using AAA. When this authorization is complete, the user has been double authenticated, and can access the network according to per-user network privileges.

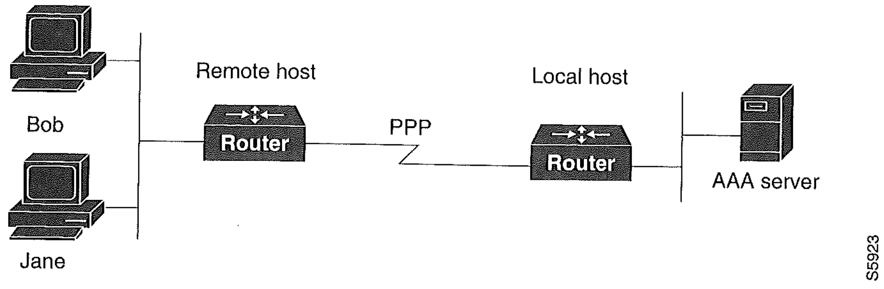
The system administrator determines what network privileges remote users will have after each stage of authentication by configuring appropriate parameters on a security server. To use double authentication, the user must activate it by issuing the **access-profile** command.

Caution Double authentication can cause certain undesirable events if multiple hosts share a PPP connection to a network access server, as shown in Figure 4.

First, if a user, Bob, initiates a PPP session and activates double authentication at the network access server (per Figure 4), any other user will automatically have the same network privileges as Bob until Bob's PPP session expires. This happens because Bob's authorization profile is applied to the network access server's interface during the PPP session and any PPP traffic from other users will use the PPP session Bob established.

Second, if Bob initiates a PPP session and activates double authentication, and then—before Bob's PPP session has expired—another user, Jane, executes the **access-profile** command (or, if Jane Telnets to the network access server and **autocommand access-profile** is executed), a reauthorization will occur and Jane's authorization profile will be applied to the interface—replacing Bob's profile. This can disrupt or halt Bob's PPP traffic, or grant Bob additional authorization privileges Bob should not have.

Figure 4 Possibly Risky Topology: Multiple Hosts Share a PPP Connection to a Network Access Server



Configuring Double Authentication

To configure double authentication, you must complete the following steps:

1. Enable AAA by using the **aaa-new model** global configuration command. For more information about enabling AAA, refer to the "AAA Overview" chapter.
2. Use the **aaa authentication** command to configure your network access server to use login and PPP authentication method lists, then apply those method lists to the appropriate lines or interfaces.
3. Use the **aaa authorization** command to configure AAA network authorization at login. For more information about configuring network authorization, refer to the "Configuring Authorization" chapter.
4. Configure security protocol parameters (for example, RADIUS or TACACS+). For more information about RADIUS, refer to the "Configuring RADIUS" chapter. For more information about TACACS+, refer to the "Configuring TACACS+" chapter.
5. Use access control list AV pairs on the security server that the user can connect to the local host only by establishing a Telnet connection.
6. (Optional) Configure the **access-profile** command as an autocommand. If you configure the autocommand, remote users will not have to manually enter the **access-profile** command to access authorized rights associated with their personal user profile. To learn about configuring autocommands, refer to the **autocommand** command in the *Cisco IOS Dial Services Command Reference.*

Note If the **access-profile** command is configured as an autocommand, users will still have to Telnet to the local host and log in to complete double authentication.

Follow these rules when creating the user-specific authorization statements (These rules relate to the default behavior of the **access-profile** command):

- Use valid AV pairs when configuring access control list AV pairs on the security server. For a list of valid AV pairs, refer to the "Authentication Commands" chapter in the *Cisco IOS Security Command Reference.*
- If you want remote users to use the interface's existing authorization (that which existed prior to the second stage authentication/authorization), but you want them to have different access control lists (ACLs), you should specify *only* ACL AV pairs in the user-specific authorization definition. This might be desirable if you set up a default authorization profile to apply to the remote host, but want to apply specific ACLs to specific users.

- When these user-specific authorization statements are later applied to the interface, they can either be *added to* the existing interface configuration or they can *replace* the existing interface configuration—depending on which form of the **access-profile** command is used to authorize the user. You should understand how the **access-profile** command works before configuring the authorization statements.
- If you will be using ISDN or Multilink PPP, you must also configure virtual templates at the local host.

To troubleshoot double authentication, use the **debug aaa per-user** debug command. For more information about this command, refer to the *Cisco IOS Debug Command Reference.*

Accessing the User Profile After Double Authentication

In double authentication, when a remote user establishes a PPP link to the local host using the local host name, the remote host is CHAP (or PAP) authenticated. After CHAP (or PAP) authentication, PPP negotiates with AAA to assign network access privileges associated with the remote host to the user. (We suggest that privileges at this stage be restricted to allow the user to connect to the local host only by establishing a Telnet connection.)

When the user needs to initiate the second phase of double authentication, establishing a Telnet connection to the local host, the user enters a personal username and password (different from the CHAP or PAP username and password). This action causes AAA reauthentication to occur according to the personal username/password. The initial rights associated with the local host, though, are still in place. By using the **access-profile** command, the rights associated with the local host are replaced by or merged with those defined for the user in the user's profile.

Use the following command in EXEC configuration mode:

Command	Purpose
`access-profile [merge	replace] [ignore-sanity-checks]`	Accesses the rights associated for the user after double authentication.

If you configured the **access-profile** command to be executed as an autocommand, it will be executed automatically after the remote user logs in.

Enabling Automated Double Authentication

You can make the double authentication process easier for users by implementing automated double authentication. Automated double authentication provides all of the security benefits of double authentication, but offers a simpler, more user-friendly interface for remote users. With double authentication, a second level of user authentication is achieved when the user Telnets to the network access server or router and enters a username and password. With automated double authentication, the user does not have to Telnet to the network access server; instead the user responds to a dialog box that requests a username and password or personal identification number (PIN). To use the automated double authentication feature, the remote user hosts must be running a companion client application. As of Cisco IOS Release 12.0, the only client application software available is the Glacier Bay application server software for PCs.

Note Automated double authentication, like the existing double authentication feature, is for Multilink PPP ISDN connections only. Automated double authentication cannot be used with other protocols such as X.25 or SLIP.

Automated double authentication is an enhancement to the existing double authentication feature. To configure automated double authentication, you must first configure double authentication by completing the following steps:

1. Enable AAA by using the **aaa-new model** global configuration command. For more information about enabling AAA, refer to the "AAA Overview" chapter.
2. Use the **aaa authentication** command to configure your network access server to use login and PPP authentication method lists, then apply those method lists to the appropriate lines or interfaces.
3. Use the **aaa authorization** command to configure AAA network authorization at login. For more information about configuring network authorization, refer to the "Configuring Authorization" chapter.
4. Configure security protocol parameters (for example, RADIUS or TACACS+). For more information about RADIUS, refer to the "Configuring RADIUS" chapter. For more information about TACACS+, refer to the "Configuring TACACS+" chapter.
5. Use access control list AV pairs on the security server that the user can connect to the local host only by establishing a Telnet connection.
6. (Optional) Configure the **access-profile** command as an autocommand. If you configure the autocommand, remote users will not have to manually enter the **access-profile** command to access authorized rights associated with their personal user profile. To learn about configuring autocommands, refer to the **autocommand** command in the *Cisco IOS Dial Services Command Reference*.

Note If the **access-profile** command is configured as an autocommand, users will still have to Telnet to the local host and log in to complete double authentication.

Follow these rules when creating the user-specific authorization statements (These rules relate to the default behavior of the **access-profile** command):

- Use valid AV pairs when configuring access control list AV pairs on the security server. For a list of valid AV pairs, refer to the "Authentication Commands" chapter in the *Cisco IOS Security Command Reference*.
- If you want remote users to use the interface's existing authorization (that which existed prior to the second stage authentication/authorization), but you want them to have different access control lists (ACLs), you should specify *only* ACL AV pairs in the user-specific authorization definition. This might be desirable if you set up a default authorization profile to apply to the remote host, but want to apply specific ACLs to specific users.
- When these user-specific authorization statements are later applied to the interface, they can either be *added to* the existing interface configuration, or *replace* the existing interface configuration—depending on which form of the **access-profile** command is used to authorize the user. You should understand how the **access-profile** command works before configuring the authorization statements.
- If you will be using ISDN or Multilink PPP, you must also configure virtual templates at the local host.

To troubleshoot double authentication, use the **debug aaa per-user** debug command. For more information about this command, refer to the *Cisco IOS Debug Command Reference*.

After you have configured double authentication, you are ready to configure the automation enhancement.

To configure automated double authentication, use the following commands, starting in global configuration mode:

Step	Command	Purpose
1	`ip trigger-authentication [timeout` *`seconds`*`] [port` *`number`*`]`	Enables automation of double authentication.

2	`interface bri` *`number`*
	or
	`interface serial` *`number`*`:23`
		Selects an ISDN BRI or ISDN PRI interface and enter the interface configuration mode.

3	`ip trigger-authentication`	Applies automated double authentication to the interface.

To troubleshoot automated double authentication, use the following commands in privileged EXEC mode:

Step	Command	Purpose
1	`show ip trigger-authentication`	Displays the list of remote hosts for which automated double authentication has been attempted (successfully or unsuccessfully).
2	`clear ip trigger-authentication`	Clears the list of remote hosts for which automated double authentication has been attempted. (This clears the table displayed by the **show ip trigger-authentication** command.)
3	`debug ip trigger-authentication`	Displays **debug** output related to automated double authentication.

Non-AAA Authentication Methods

This section discusses the following non-AAA authentication tasks:

- Configuring Line Password Protection
- Establishing Username Authentication
- Enabling CHAP or PAP Authentication
- Using MS-CHAP

Configuring Line Password Protection

You can provide access control on a terminal line by entering the password and establishing password checking. To do so, use the following commands in line configuration mode:

Step	Command	Purpose
1	`password` *`password`*	Assigns a password to a terminal or other device on a line.
2	`login`	Enables password checking at login.

The password checker is case sensitive and can include spaces; for example, the password "Secret" is different from the password "secret," and "two words" is an acceptable password.

You can disable line password verification by disabling password checking. To do so, use the following command in line configuration mode:

Command	Purpose
no login	Disables password checking or allow access to a line without password verification.

If you configure line password protection and then configure TACACS or extended TACACS, the TACACS username and password take precedence over line passwords. If you have not yet implemented a security policy, we recommend that you use AAA.

Note The **login** command only changes username and privilege level but it does not execute a shell; therefore autocommands will not be executed. To execute autocommands under this circumstance, you need to establish a Telnet session back into the router (loop-back). Make sure that the router has been configured for secure Telnet sessions if you choose to implement autocommands this way.

Establishing Username Authentication

You can create a username-based authentication system, which is useful in the following situations:

- To provide a TACACS-like username and encrypted password-authentication system for networks that cannot support TACACS
- To provide special-case logins: for example, access list verification, no password verification, autocommand execution at login, and "no escape" situations

To establish username authentication, use the following commands in global configuration mode as needed for your system configuration:

Step	Command	Purpose
1	**username** *name* [**nopassword** | **password** *password* | **password** *encryption-type encrypted password*] **username** *name* [**access-class** *number*]	Establishes username authentication with encrypted passwords. or (Optional) Establishes username authentication by access list.
2	**username** *name* [**privilege** *level*]	(Optional) Sets the privilege level for the user.
3	**username** *name* [**autocommand** *command*]	(Optional) Specifies a command to automatically execute.
4	**username** *name* [**noescape**] [**nohangup**]	(Optional) Sets a "no escape" login environment.

The keyword **noescape** prevents users from using escape characters on the hosts to which they are connected. The **nohangup** feature does not disconnect after using the autocommand.

Caution Passwords will be displayed in clear text in your configuration unless you enable the **service password-encryption** command. For more information about the **service password-encryption** command, refer to the "Passwords and Privileges Commands" chapter in the *Cisco IOS Security Command Reference*.

Enabling CHAP or PAP Authentication

One of the most common transport protocols used in Internet Service Providers' (ISPs') dial solutions is the Point-to-Point Protocol (PPP). Traditionally, remote users dial in to an access server to initiate a PPP session. After PPP has been negotiated, remote users are connected to the ISP network and to the Internet.

Because ISPs want only customers to connect to their access servers, remote users are required to authenticate to the access server before they can start up a PPP session. Normally, a remote user authenticates by typing in a username and password when prompted by the access server. Although this is a workable solution, it is difficult to administer and awkward for the remote user.

A better solution is to use the authentication protocols built into PPP. In this case, the remote user dials in to the access server and starts up a minimal subset of PPP with the access server. This does not give the remote user access to the ISP's network—it merely allows the access server to talk to the remote device.

PPP currently supports two authentication protocols: Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP). Both are specified in RFC 1334 and are supported on synchronous and asynchronous interfaces. Authentication via PAP or CHAP is equivalent to typing in a username and password when prompted by the server. CHAP is considered to be more secure because the remote user's password is never sent across the connection.

PPP (with or without PAP or CHAP authentication) is also supported in dialout solutions. An access server utilizes a dialout feature when it initiates a call to a remote device and attempts to start up a transport protocol such as PPP.

See the "Configuring Interfaces" chapter in the *Cisco IOS Configuration Fundamentals Configuration Guide* for more information about CHAP and PAP.

Note To use CHAP or PAP, you must be running PPP encapsulation.

When CHAP is enabled on an interface and a remote device attempts to connect to it, the access server sends a CHAP packet to the remote device. The CHAP packet requests or "challenges" the remote device to respond. The challenge packet consists of an ID, a random number, and the host name of the local router.

When the remote device receives the challenge packet, it concatenates the ID, the remote device's password, and the random number, and then encrypts all of it using the remote device's password. The remote device sends the results back to the access server, along with the name associated with the password used in the encryption process.

When the access server receives the response, it uses the name it received to retrieve a password stored in its user database. The retrieved password should be the same password the remote device used in its encryption process. The access server then encrypts the concatenated information with the newly-retrieved password—if the result matches the result sent in the response packet, authentication succeeds.

The benefit of using CHAP authentication is that the remote device's password is never transmitted in clear text. This prevents other devices from stealing it and gaining illegal access to the ISP's network.

CHAP transactions occur only at the time a link is established. The access server does not request a password during the rest of the call. (The local device can, however, respond to such requests from other devices during a call.)

When PAP is enabled, the remote router attempting to connect to the access server is required to send an authentication request. If the username and password specified in the authentication request are accepted, the Cisco IOS software sends an authentication acknowledgment.

After you have enabled CHAP or PAP, the access server will require authentication from remote devices dialing in to the access server. If the remote device does not support the enabled protocol, the call will be dropped.

To use CHAP or PAP, you must perform the following tasks:

1. Enable PPP encapsulation.
2. Enable CHAP or PAP on the interface.
3. For CHAP, configure host name authentication and the secret or password for each remote system with which authentication is required.

This section includes the following sections:

- Enabling PPP Encapsulation
- Enabling PAP or CHAP
- Inbound and Outbound Authentication
- Enabling Outbound PAP Authentication
- Creating a Common CHAP Password
- Refusing CHAP Authentication Requests
- Delaying CHAP Authentication Until Peer Authenticates

Enabling PPP Encapsulation

To enable PPP encapsulation, use the following command in interface configuration mode:

Command	Purpose
`encapsulation ppp`	Enables PPP on an interface.

Enabling PAP or CHAP

To enable CHAP or PAP authentication on an interface configured for PPP encapsulation, use the following command in interface configuration mode:

Command	Purpose
ppp authentication {*protocol1* [*protocol2*...]} [**if-needed**] {**default** | *list-name*} [**callin**] [**one-time**]	Defines the authentication protocols supported and the order in which they are used. In this command, *protocol1, protocol2* represent the following protocols: CHAP, MS-CHAP, and PAP. PPP authentication is attempted first using the first authentication method, which is *protocol1*. If *protocol1* is unable to establish authentication, the next configured protocol is used to negotiate authentication.

If you configure **ppp authentication chap** on an interface, all incoming calls on that interface that initiate a PPP connection will have to be authenticated using CHAP; likewise, if you configure **ppp authentication pap**, all incoming calls that start a PPP connection will have to be authenticated via PAP. If you configure **ppp authentication chap pap**, the access server will attempt to authenticate all incoming calls that start a PPP session with CHAP. If the remote device does not support CHAP, the access server will try to authenticate the call using PAP. If the remote device does not support either CHAP or PAP, authentication will fail and the call will be dropped. If you configure **ppp authentication pap chap**, the access server will attempt to authenticate all incoming calls that start a PPP session with PAP. If the remote device does not support PAP, the access server will try to authenticate the call using CHAP. If the remote device does not support either protocol, authentication will fail and the call will be dropped. If you configure the **ppp authentication** command with the **callin** keyword, the access server will only authenticate the remote device if the remote device initiated the call.

Authentication method lists and the **one-time** keyword are only available if you have enabled AAA—they will not be available if you are using TACACS or extended TACACS. If you specify the name of an authentication method list with the **ppp authentication** command, PPP will attempt to authenticate the connection using the methods defined in the specified method list. If AAA is enabled and no method list is defined by name, PPP will attempt to authenticate the connection using the methods defined as the default. The **ppp authentication** command with the **one-time** keyword enables support for one-time passwords during authentication.

The **if-needed** keyword is only available if you are using TACACS or extended TACACS. The **ppp authentication** command with the **if-needed** keyword means that PPP will only authenticate the remote device via PAP or CHAP if they have not yet authenticated during the life of the current call. If the remote device authenticated via a standard login procedure and initiated PPP from the EXEC prompt, PPP will not authenticate via CHAP if **ppp authentication chap if-needed** is configured on the interface.

Caution If you use a *list-name* that has not been configured with the **aaa authentication ppp** command, you disable PPP on the line.

For information about adding a **username** entry for each remote system from which the local router or access server requires authentication, see the "Establishing Username Authentication" section.

Inbound and Outbound Authentication

PPP supports two-way authentication. Normally, when a remote device dials in to an access server, the access server requests that the remote device prove that it is allowed access. This is known as inbound authentication. At the same time, the remote device can also request that the access server prove that it is who it says it is. This is known as outbound authentication. An access server also does outbound authentication when it initiates a call to a remote device.

Enabling Outbound PAP Authentication

To enable outbound PAP authentication, use the following command in interface configuration mode:

Command	Purpose
`ppp pap sent-username` *username* `password` *password*	Enables outbound PAP authentication.

The access server uses the username and password specified by the **ppp pap sent-username** command to authenticate itself whenever it initiates a call to a remote device or when it has to respond to a remote device's request for outbound authentication.

Creating a Common CHAP Password

For remote CHAP authentication only, you can configure your router to create a common CHAP secret password to use in response to challenges from an unknown peer; for example, if your router calls a rotary of routers (either from another vendor, or running an older version of the Cisco IOS software) to which a new (that is, unknown) router has been added. The **ppp chap password** command allows you to replace several username and password configuration commands with a single copy of this command on any dialer interface or asynchronous group interface.

To enable a router calling a collection of routers to configure a common CHAP secret password, use the following command in interface configuration mode:

Command	Purpose
`ppp chap password` *`secret`*	Enables a router calling a collection of routers to configure a common CHAP secret password.

Refusing CHAP Authentication Requests

To refuse CHAP authentication from peers requesting it, meaning that CHAP authentication is disabled for all calls, use the following command in interface configuration mode:

Command	Purpose
`ppp chap refuse [callin]`	Refuses CHAP authentication from peers requesting CHAP authentication.

If the **callin** keyword is used, the router will refuse to answer CHAP authentication challenges received from the peer, but will still require the peer to answer any CHAP challenges the router sends.

If outbound PAP has been enabled (using the **ppp pap sent-username** command), PAP will be suggested as the authentication method in the refusal packet.

Delaying CHAP Authentication Until Peer Authenticates

To specify that the router will not authenticate to a peer requesting CHAP authentication until after the peer has authenticated itself to the router, use the following command in interface configuration mode:

Command	Purpose
`ppp chap wait` *`secret`*	Configures the router to delay CHAP authentication until after the peer has authenticated itself to the router.

This command (which is the default) specifies that the router will not authenticate to a peer requesting CHAP authentication until the peer has authenticated itself to the router. The **no ppp chap wait** command specifies that the router will respond immediately to an authentication challenge.

Using MS-CHAP

Microsoft Challenge Handshake Authentication Protocol (MS-CHAP) is the Microsoft version of CHAP and is an extension to RFC 1994. Like the standard version of CHAP, MS-CHAP is used for PPP authentication; in this case, authentication occurs between a PC using Microsoft Windows NT or Microsoft Windows 95 and a Cisco router or access server acting as a network access server.

MS-CHAP differs from the standard CHAP as follows:

- MS-CHAP is enabled by negotiating CHAP Algorithm 0x80 in LCP option 3, Authentication Protocol.
- The MS-CHAP Response packet is in a format designed to be compatible with Microsoft Windows NT 3.5 and 3.51, Microsoft Windows 95, and Microsoft LAN Manager 2.*x*. This format does not require the authenticator to store a clear or reversibly encrypted password.
- MS-CHAP provides an authenticator-controlled authentication retry mechanism.

- MS-CHAP provides an authenticator-controlled change password mechanism.
- MS-CHAP defines a set of "reason-for failure" codes returned in the Failure packet message field.

Depending on the security protocols you have implemented, PPP authentication using MS-CHAP can be used with or without AAA security services. If you have enabled AAA, PPP authentication using MS-CHAP can be used in conjunction with both TACACS+ and RADIUS. Two new vendor-specific RADIUS attributes (IETF Attribute 26) were added to enable RADIUS to support MS-CHAP. These new attributes are listed in Table 9.

Table 9 Vendor-Specific RADIUS Attributes for MS-CHAP

Vendor-ID Number	Vendor-Type Number	Vendor-Proprietary Attribute	Description
311	11	MSCHAP-Challenge	Contains the challenge sent by a network access server to an MS-CHAP user. It can be used in both Access-Request and Access-Challenge packets.
311	1	MSCHAP-Response	Contains the response value provided by a PPP MS-CHAP user in response to the challenge. It is only used in Access-Request packets. This attribute is identical to the PPP CHAP Identifier.

Use the following commands in interface configuration mode:

Step	Command	Purpose
1	`encapsulation ppp`	Enables PPP encapsulation.
2	`ppp authentication ms-chap [if-needed] [`*`list-name`*`	default] [callin] [one-time]`	Defines PPP authentication using MS-CHAP.

If you configure **ppp authentication ms-chap** on an interface, all incoming calls on that interface that initiate a PPP connection will have to be authenticated using MS-CHAP. If you configure the **ppp authentication** command with the **callin** keyword, the access server will only authenticate the remote device if the remote device initiated the call.

Authentication method lists and the **one-time** keyword are only available if you have enabled AAA—they will not be available if you are using TACACS or extended TACACS. If you specify the name of an authentication method list with the **ppp authentication** command, PPP will attempt to authenticate the connection using the methods defined in the specified method list. If AAA is enabled and no method list is defined by name, PPP will attempt to authenticate the connection using the methods defined as the default. The **ppp authentication** command with the **one-time** keyword enables support for one-time passwords during authentication.

The **if-needed** keyword is only available if you are using TACACS or extended TACACS. The **ppp authentication** command with the **if-needed** keyword means that PPP will only authenticate the remote device via MS-CHAP if that device has not yet authenticated during the life of the current call. If the remote device authenticated through a standard login procedure and initiated PPP from the EXEC prompt, PPP will not authenticate through MS-CHAP if **ppp authentication chap if-needed** is configured.

Note If PPP authentication using MS-CHAP is used with username authentication, you must include the MS-CHAP secret in the local username/password database. For more information about username authentication, refer to the "Establish Username Authentication" section.

Authentication Examples

The following sections provide authentication configuration examples:

- RADIUS Authentication Examples
- TACACS+ Authentication Examples
- Kerberos Authentication Examples
- AAA Scalability Example
- Login and Failed Banner Examples
- Double Authentication Examples
- Automated Double Authentication Example
- MS-CHAP Example

RADIUS Authentication Examples

This section provides two sample configurations using RADIUS.

The following example shows how to configure the router to authenticate and authorize using RADIUS:

```
aaa authentication login radius-login group radius local
aaa authentication ppp radius-ppp if-needed group radius
aaa authorization exec default group radius if-authenticated
aaa authorization network default group radius
line 3
login authentication radius-login
interface serial 0
ppp authentication radius-ppp
```

The lines in this sample RADIUS authentication and authorization configuration are defined as follows:

- The **aaa authentication login radius-login group radius local** command configures the router to use RADIUS for authentication at the login prompt. If RADIUS returns an error, the user is authenticated using the local database.
- The **aaa authentication ppp radius-ppp if-needed group radius** command configures the Cisco IOS software to use PPP authentication using CHAP or PAP if the user has not already logged in. If the EXEC facility has authenticated the user, PPP authentication is not performed.
- The **aaa authorization exec default group radius if-authenticated** command queries the RADIUS database for information that is used during EXEC authorization, such as autocommands and privilege levels, but only provides authorization if the user has successfully authenticated.
- The **aaa authorization network default group radius** command queries RADIUS for network authorization, address assignment, and other access lists.

- The **login authentication radius-login** command enables the radius-login method list for line 3.
- The **ppp authentication radius-ppp** command enables the radius-ppp method list for serial interface 0.

The following example shows how to configure the router to prompt for and verify a username and password, authorize the user's EXEC level, and specify it as the method of authorization for privilege level 2. In this example, if a local username is entered at the username prompt, that username is used for authentication.

If the user is authenticated using the local database, EXEC authorization using RADIUS will fail because no data is saved from the RADIUS authentication. The method list also uses the local database to find an autocommand. If there is no autocommand, the user becomes the EXEC user. If the user then attempts to issue commands that are set at privilege level 2, TACACS+ is used to attempt to authorize the command.

```
aaa authentication login default group radius local
aaa authorization exec default group radius local
aaa authorization command 2 default group tacacs+ if-authenticated
```

The lines in this sample RADIUS authentication and authorization configuration are defined as follows:

- The **aaa authentication login default group radius local** command specifies that the username and password are verified by RADIUS or, if RADIUS is not responding, by the router's local user database.
- The **aaa authorization exec default group radius local** command specifies that RADIUS authentication information be used to set the user's EXEC level if the user authenticates with RADIUS. If no RADIUS information is used, this command specifies that the local user database be used for EXEC authorization.
- The **aaa authorization command 2 default group tacacs+ if-authenticated** command specifies TACACS+ authorization for commands set at privilege level 2, if the user has already successfully authenticated.

TACACS+ Authentication Examples

The following example shows how to configure TACACS+ as the security protocol to be used for PPP authentication:

```
aaa new-model
aaa authentication ppp test group tacacs+ local
interface serial 0
ppp authentication chap pap test
tacacs-server host 10.1.2.3
tacacs-server key goaway
```

The lines in this sample TACACS+ authentication configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "test," to be used on serial interfaces running PPP. The keywords **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **interface** command selects the line.
- The **ppp authentication** command applies the test method list to this line.

- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.1.2.3.
- The **tacacs-server key** command defines the shared encryption key to be "goaway."

The following example shows how to configure AAA authentication for PPP:

```
aaa authentication ppp default if-needed group tacacs+ local
```

In this example, the keyword **default** means that PPP authentication is applied by default to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP is not necessary and can be skipped. If authentication is needed, the keywords **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.

The following example shows how to create the same authentication algorithm for PAP, but it calls the method list "MIS-access" instead of "default":

```
aaa authentication ppp MIS-access if-needed group tacacs+ local
interface serial 0
ppp authentication pap MIS-access
```

In this example, because the list does not apply to any interfaces (unlike the default list, which applies automatically to all interfaces), the administrator must select interfaces to which this authentication scheme should apply by using the **interface** command. The administrator must then apply this method list to those interfaces by using the **ppp authentication** command.

Kerberos Authentication Examples

To specify Kerberos as the login authentication method, use the following command:

```
aaa authentication login default krb5
```

To specify Kerberos authentication for PPP, use the following command:

```
aaa authentication ppp default krb5
```

AAA Scalability Example

The following example shows a general security configuration using AAA with RADIUS as the security protocol. In this example, the network access server is configured to allocate 16 background processes to handle AAA requests for PPP.

```
aaa new-model
radius-server host alcatraz
radius-server key myRaDiUSpassWoRd
radius-server configure-nas
username root password ALongPassword
aaa authentication ppp dialins group radius local
aaa authentication login admins local
aaa authorization network default group radius local
aaa accounting network default start-stop group radius
aaa processes 16
line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem dialin
interface group-async 1
 group-range 1 16
 encapsulation ppp
 ppp authentication pap dialins
```

The lines in this sample RADIUS AAA configuration are defined as follows:

- The **aaa new-model** command enables AAA network security services.
- The **radius-server host** command defines the name of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **radius-server configure-nas** command defines that the Cisco router or access server will query the RADIUS server for static routes and IP pool definitions when the device first starts up.
- The **username** command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list "dialins," which specifies that RADIUS authentication, then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.
- The **aaa authentication login admins local** command defines another method list, "admins," for login authentication.
- The **aaa authorization network default group radius local** command is used to assign an address and other network parameters to the RADIUS user.
- The **aaa accounting network default start-stop group radius** command tracks PPP usage.
- The **aaa processes** command allocates 16 background processes to handle AAA requests for PPP.
- The **line** command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.
- The **autoselect ppp** command configures the Cisco IOS software to allow a PPP session to start up automatically on these selected lines.

- The **autoselect during-login** command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.
- The **login authentication admins** command applies the "admins" method list for login authentication.
- The **modem dialin** command configures modems attached to the selected lines to only accept incoming calls.
- The **interface group-async** command selects and defines an asynchronous interface group.
- The **group-range** command defines the member asynchronous interfaces in the interface group.
- The **encapsulation ppp** command sets PPP as the encapsulation method used on the specified interfaces.
- The **ppp authentication pap dialins** command applies the "dialins" method list to the specified interfaces.

Login and Failed Banner Examples

The following example shows how to configure a login banner (in this case, the phrase "Unauthorized Access Prohibited") that will be displayed when a user logs in to the system. The asterisk (*) is used as the delimiting character. (RADIUS is specified as the default login authentication method.)

```
aaa new-model
aaa authentication banner *Unauthorized Access Prohibited*
aaa authentication login default group radius
```

This configuration produces the following login banner:

```
Unauthorized Access Prohibited
Username:
```

The following example shows how to additionally configure a failed login banner (in this case, the phrase "Failed login. Try again.") that will be displayed when a user tries to log in to the system and fails. The asterisk (*) is used as the delimiting character. (RADIUS is specified as the default login authentication method.)

```
aaa new-model
aaa authentication banner *Unauthorized Access Prohibited*
aaa authentication fail-message *Failed login. Try again.*
aaa authentication login default group radius
```

This configuration produces the following login and failed login banner:

```
Unauthorized Access Prohibited
Username:
Password:
Failed login. Try again.
```

Double Authentication Examples

The examples in this section illustrate possible configurations to be used with double authentication. Your configurations could differ significantly, depending on your network and security requirements.

This section includes the following examples:

- Configuration of the Local Host for AAA with Double Authentication Examples
- Configuration of the AAA Server for First-Stage (PPP) Authentication/Authorization Example
- Configuration of the AAA Server for Second-Stage (Per-User) Authentication/Authorization Examples
- Complete Configuration with TACACS+ Example

Note These configuration examples include specific IP addresses and other specific information. This information is for illustration purposes only: your configuration will use different IP addresses, different usernames and passwords, and different authorization statements.

Configuration of the Local Host for AAA with Double Authentication Examples

These two examples show how to configure a local host to use AAA for PPP and login authentication, and for network and EXEC authorization. One example is shown for RADIUS and one example for TACACS+.

In both examples, the first three lines configure AAA, with a specific server as the AAA server. The next two lines configure AAA for PPP and login authentication, and the last two lines configure network and EXEC authorization. The last line is necessary only if the **access-profile** command will be executed as an autocommand.

The following example shows router configuration with a RADIUS AAA server.

```
aaa new-model
radius-server host secureserver
radius-server key myradiuskey
aaa authentication ppp default group radius
aaa authentication login default group radius
aaa authorization network default group radius
aaa authorization exec default group radius
```

The following example shows router configuration with a TACACS+ server.

```
aaa new-model
tacacs-server host security
tacacs-server key mytacacskey
aaa authentication ppp default group tacacs+
aaa authentication login default group tacacs+
aaa authorization network default group tacacs+
aaa authorization exec default group tacacs+
```

Configuration of the AAA Server for First-Stage (PPP) Authentication/Authorization Example

This example shows a configuration on the AAA server. A partial sample AAA configuration is shown for RADIUS.

TACACS+ servers can be configured similarly. (See the "Complete Configuration with TACACS+ Example" section later in this document.)

This example defines authentication/authorization for a remote host named "hostx" that will be authenticated by CHAP in the first stage of double authentication. Note that the ACL AV pair limits the remote host to Telnet connections to the local host. The local host has the IP address 10.0.0.2.

The following example shows a partial AAA server configuration for RADIUS.

```
hostx     Password = "welcome"
          User-Service-Type = Framed-User,
          Framed-Protocol = PPP,
          cisco-avpair = "lcp:interface-config=ip unnumbered ethernet 0",
          cisco-avpair = "ip:inacl#3=permit tcp any 172.21.114.0 0.0.0.255 eq telnet",
          cisco-avpair = "ip:inacl#4=deny icmp any any",
          cisco-avpair = "ip:route#5=55.0.0.0 255.0.0.0",
          cisco-avpair = "ip:route#6=66.0.0.0 255.0.0.0",
          cisco-avpair = "ipx:inacl#3=deny any"
```

Configuration of the AAA Server for Second-Stage (Per-User) Authentication/Authorization Examples

This section contains partial sample AAA configurations on a RADIUS server. These configurations define authentication/authorization for a user (Pat) with the username "patuser," who will be user-authenticated in the second stage of double authentication.

TACACS+ servers can be configured similarly. (See the "Complete Configuration with TACACS+ Example" section later in this document.)

Three examples show sample RADIUS AAA configurations that could be used with each of the three forms of the **access-profile** command.

The first example shows a partial sample AAA configuration that works with the default form (no keywords) of the **access-profile** command. Note that only ACL AV pairs are defined. This example also sets up the **access-profile** command as an autocommand.

```
patuser   Password = "welcome"
          User-Service-Type = Shell-User,
          cisco-avpair = "shell:autocmd=access-profile"
          User-Service-Type = Framed-User,
          Framed-Protocol = PPP,
          cisco-avpair = "ip:inacl#3=permit tcp any host 10.0.0.2 eq telnet",
          cisco-avpair = "ip:inacl#4=deny icmp any any"
```

The second example shows a partial sample AAA configuration that works with the **access-profile merge** form of the **access-profile** command. This example also sets up the **access-profile merge** command as an autocommand.

```
patuser   Password = "welcome"
          User-Service-Type = Shell-User,
          cisco-avpair = "shell:autocmd=access-profile merge"
          User-Service-Type = Framed-User,
          Framed-Protocol = PPP,
          cisco-avpair = "ip:inacl#3=permit tcp any any"
          cisco-avpair = "ip:route=10.0.0.0 255.255.0.0",
          cisco-avpair = "ip:route=10.1.0.0 255.255.0.0",
          cisco-avpair = "ip:route=10.2.0.0 255.255.0.0"
```

The third example shows a partial sample AAA configuration that works with the **access-profile replace** form of the **access-profile** command. This example also sets up the **access-profile replace** command as an autocommand.

```
patuser   Password = "welcome"
          User-Service-Type = Shell-User,
          cisco-avpair = "shell:autocmd=access-profile replace"
          User-Service-Type = Framed-User,
          Framed-Protocol = PPP,
          cisco-avpair = "ip:inacl#3=permit tcp any any",
          cisco-avpair = "ip:inacl#4=permit icmp any any",
          cisco-avpair = "ip:route=10.10.0.0 255.255.0.0",
          cisco-avpair = "ip:route=10.11.0.0 255.255.0.0",
          cisco-avpair = "ip:route=10.12.0.0 255.255.0.0"
```

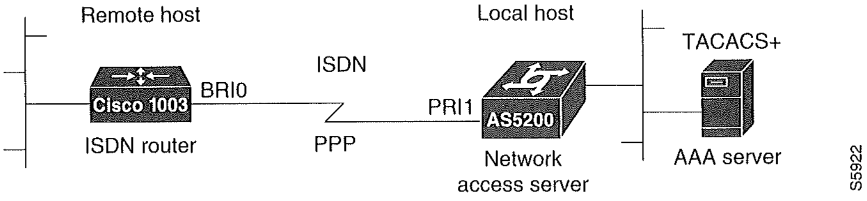
Complete Configuration with TACACS+ Example

This example shows TACACS+ authorization profile configurations both for the remote host (used in the first stage of double authentication) and for specific users (used in the second stage of double authentication). This TACACS+ example contains approximately the same configuration information as shown in the previous RADIUS examples.

This sample configuration shows authentication/authorization profiles on the TACACS+ server for the remote host "hostx" and for three users, with the usernames "pat_default," "pat_merge," and "pat_replace." The configurations for these three usernames illustrate different configurations that correspond to the three different forms of the **access-profile** command. The three user configurations also illustrate setting up the autocommand for each form of the **access-profile** command.

Figure 5 shows the topology. The example that follows the figure shows a TACACS+ configuration file.

Figure 5 Example Topology for Double Authentication



This sample configuration shows authentication/authorization profiles on the TACACS+ server for the remote host "hostx" and for three users, with the usernames "pat_default," "pat_merge," and "pat_replace."

```
key = "mytacacskey"

default authorization = permit


#-----------------------------Remote Host (BRI)------------------------
#
# This allows the remote host to be authenticated by the local host
# during fist-stage authentication, and provides the remote host
# authorization profile.
#
#-----------------------------------------------------------------------
```

```
user = hostx
{
    login = cleartext "welcome"
    chap = cleartext "welcome"


    service = ppp protocol = lcp {
                interface-config="ip unnumbered ethernet 0"
    }

    service = ppp protocol = ip {
            # It is important to have the hash sign and some string after
            # it. This indicates to the NAS that you have a per-user
            # config.

            inacl#3="permit tcp any 172.21.114.0 0.0.0.255 eq telnet"
            inacl#4="deny icmp any any"

            route#5="55.0.0.0 255.0.0.0"
            route#6="66.0.0.0 255.0.0.0"
    }

    service = ppp protocol = ipx {
            # see previous comment about the hash sign and string, in protocol = ip
            inacl#3="deny any"
    }


}


#------------------- "access-profile" default user "only acls" ------------------
#
# Without arguments, access-profile removes any access-lists it can find
# in the old configuration (both per-user and per-interface), and makes sure
# that the new profile contains ONLY access-list definitions.
#
#--------------------------------------------------------------------------------

user = pat_default
{
        login = cleartext "welcome"
        chap = cleartext "welcome"

        service = exec

        {
                # This is the autocommand that executes when pat_default logs in.
                autocmd = "access-profile"
        }

        service = ppp protocol = ip {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IP
                # access-lists (not even the ones installed prior to
                # this)!

                inacl#3="permit tcp any host 10.0.0.2 eq telnet"
                inacl#4="deny icmp any any"
        }


        service = ppp protocol = ipx {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IPX
                # access-lists (not even the ones installed prior to
                # this)!
        }


}


#--------------------- "access-profile merge" user ---------------------------
#
# With the 'merge' option, first all old access-lists are removed (as before),
# but then (almost) all AV pairs are uploaded and installed. This will allow
# for uploading any custom static routes, sap-filters, and so on, that the user
# may need in his or her profile. This needs to be used with care, as it leaves
# open the possibility of conflicting configurations.
#
#-----------------------------------------------------------------------------

user = pat_merge
{
        login = cleartext "welcome"
        chap = cleartext "welcome"


        service = exec
        {
                # This is the autocommand that executes when pat_merge logs in.
                autocmd = "access-profile merge"
        }

        service = ppp protocol = ip
        {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IP
                # access-lists (not even the ones installed prior to
                # this)!

                inacl#3="permit tcp any any"
                route#2="10.0.0.0 255.255.0.0"
                route#3="10.1.0.0 255.255.0.0"
                route#4="10.2.0.0 255.255.0.0"

        }

        service = ppp protocol = ipx
        {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IPX
                # access-lists (not even the ones installed prior to
                # this)!

        }

}


#--------------------- "access-profile replace" user ----------------------------
#
# With the 'replace' option, ALL old configuration is removed and ALL new
# configuration is installed.
#
```

```
# One caveat: access-profile checks the new configuration for address-pool and
# address AV pairs. As addresses cannot be renegotiated at this point, the
# command will fail (and complain) when it encounters such an AV pair.
# Such AV pairs are considered to be "invalid" for this context.
#-------------------------------------------------------------------------------


user = pat_replace
{
        login = cleartext "welcome"
        chap = cleartext "welcome"


        service = exec
        {
                # This is the autocommand that executes when pat_replace logs in.
                autocmd = "access-profile replace"
        }

        service = ppp protocol = ip
        {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IP
                # access-lists (not even the ones installed prior to
                # this)!

                inacl#3="permit tcp any any"
                inacl#4="permit icmp any any"


                route#2="10.10.0.0 255.255.0.0"
                route#3="10.11.0.0 255.255.0.0"
                route#4="10.12.0.0 255.255.0.0"
        }

        service = ppp protocol = ipx
        {
                # put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IPX
                # access-lists (not even the ones installed prior to
                # this)!
        }

}
```

Automated Double Authentication Example

This example shows a complete configuration file for a Cisco 2509 router with automated double authentication configured. The configuration commands that apply to automated double authentication are preceded by descriptions with a double asterisk (**).

```
Current configuration:
!
version 11.3
no service password-encryption
!
hostname myrouter
!
!
! **The following AAA commands are used to configure double authentication:
!
! **The following command enables AAA:
aaa new-model
! **The following command enables user authentication via the TACACS+ AAA server:
aaa authentication login default group tacacs+
aaa authentication login console none
! **The following command enables device authentication via the TACACS+ AAA server:
aaa authentication ppp default group tacacs+
! **The following command causes the remote user's authorization profile to be
! downloaded from the AAA server to the Cisco 2509 router when required:
aaa authorization exec default group tacacs+
! **The following command causes the remote device's authorization profile to be
! downloaded from the AAA server to the Cisco 2509 router when required:
aaa authorization network default group tacacs+
enable password mypassword
!
ip host blue 172.21.127.226
ip host green 172.21.127.218
ip host red 172.21.127.114
ip domain-name example.com
ip name-server 171.69.2.75
! **The following command globally enables automated double authentication:
ip trigger-authentication timeout 60 port 7500
isdn switch-type basic-5ess
!
!
interface Ethernet0
 ip address 172.21.127.186 255.255.255.248
 no ip route-cache
 no ip mroute-cache
 no keepalive
 ntp disable
 no cdp enable
!
interface Virtual-Template1
 ip unnumbered Ethernet0
 no ip route-cache
 no ip mroute-cache
!
interface Serial0
 ip address 172.21.127.105 255.255.255.248
 encapsulation ppp
 no ip mroute-cache
 no keepalive
 shutdown
 clockrate 2000000
 no cdp enable
!
interface Serial1
```

```
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no cdp enable
!
! **Automated double authentication occurs via the ISDN BRI interface BRI0:
interface BRI0
 ip unnumbered Ethernet0
! **The following command turns on automated double authentication at this interface:
 ip trigger-authentication
! **PPP encapsulation is required:
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 dialer idle-timeout 500
 dialer map ip 172.21.127.113 name myrouter 60074
 dialer-group 1
 no cdp enable
! **The following command specifies that device authentication occurs via PPP CHAP:
 ppp authentication chap
!
router eigrp 109
 network 172.21.0.0
 no auto-summary
!
ip default-gateway 172.21.127.185
no ip classless
ip route 172.21.127.114 255.255.255.255 172.21.127.113
! **Virtual profiles are required for double authentication to work:
virtual-profile virtual-template 1
dialer-list 1 protocol ip permit
no cdp run
! **The following command defines where the TACACS+ AAA server is:
tacacs-server host 171.69.57.35 port 1049
tacacs-server timeout 90
! **The following command defines the key to use with TACACS+ traffic (required):
tacacs-server key mytacacskey
snmp-server community public RO
!
line con 0
 exec-timeout 0 0
 login authentication console
line aux 0
 transport input all
line vty 0 4
 exec-timeout 0 0
 password lab
!
end
```

MS-CHAP Example

The following example shows how to configure a Cisco AS5200 (enabled for AAA and communication with a RADIUS security server) for PPP authentication using MS-CHAP:

```
aaa new-model
aaa authentication login admins local
aaa authentication ppp dialins group radius local
aaa authorization network default group radius local
aaa accounting network default start-stop group radius

username root password ALongPassword

radius-server host alcatraz
radius-server key myRaDiUSpassWoRd

interface group-async 1
 group-range 1 16
 encapsulation ppp
 ppp authentication ms-chap dialins

line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem dialin
```

The lines in this sample RADIUS AAA configuration are defined as follows:

- The **aaa new-model** command enables AAA network security services.
- The **aaa authentication login admins local** command defines another method list, "admins," for login authentication.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list "dialins," which specifies that RADIUS authentication then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.
- The **aaa authorization network default group radius local** command is used to assign an address and other network parameters to the RADIUS user.
- The **aaa accounting network default start-stop group radius** command tracks PPP usage.
- The **username** command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.
- The **radius-server host** command defines the name of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **interface group-async** command selects and defines an asynchronous interface group.
- The **group-range** command defines the member asynchronous interfaces in the interface group.
- The **encapsulation ppp** command sets PPP as the encapsulation method used on the specified interfaces.
- The **ppp authentication ms-chap dialins** command selects MS-CHAP as the method of PPP authentication and applies the "dialins" method list to the specified interfaces.
- The **line** command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.

- The **autoselect ppp** command configures the Cisco IOS software to allow a PPP session to start up automatically on these selected lines.
- The **autoselect during-login** command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.
- The **login authentication admins** command applies the "admins" method list for login authentication.
- The **modem dialin** command configures modems attached to the selected lines to only accept incoming calls.

Configuring Authorization

AAA authorization enables you to limit the services available to a user. When AAA authorization is enabled, the network access server uses information retrieved from the user's profile, which is located either in the local user database or on the security server, to configure the user's session. Once this is done, the user will be granted access to a requested service only if the information in the user profile allows it.

For a complete description of the authorization commands used in this chapter, refer to the "Authorization Commands" chapter in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

In This Chapter

This chapter describes the following topics and tasks:

- Named Method Lists for Authorization
- AAA Authorization Methods
- Method Lists and Server Groups
- AAA Authorization Types
- AAA Authorization Prerequisites
- AAA Authorization Configuration Task List
- Authorization Attribute-Value Pairs
- Authorization Configuration Examples

Named Method Lists for Authorization

Method lists for authorization define the ways authorization will be performed and the sequence in which these methods will be performed. A method list is simply a named list describing the authorization methods to be queried (such as RADIUS or TACACS+), in sequence. Method lists enable you to designate one or more security protocols to be used for authorization, thus ensuring a backup system in case the initial method fails. Cisco IOS software uses the first method listed to authorize users for specific network services; if that method fails to respond, the Cisco IOS software selects the next method listed in the method list. This process continues until there is successful communication with a listed authorization method, or all methods defined are exhausted.

Note The Cisco IOS software attempts authorization with the next listed method only when there is no response from the previous method. If authorization fails at any point in this cycle—meaning that the security server or local username database responds by denying the user services—the authorization process stops and no other authorization methods are attempted.

Method lists are specific to the authorization type requested:

- **Auth-proxy**—Applies specific security policies on a per-user basis. For detailed information on the authentication proxy feature, refer to the "Configuring Authentication Proxy" chapter in the "Traffic Filtering and Firewalls" section of this book.
- **Commands**—Applies to the EXEC mode commands a user issues. Command authorization attempts authorization for all EXEC mode commands, including global configuration commands, associated with a specific privilege level.
- **EXEC**—Applies to the attributes associated with a user EXEC terminal session.
- **Network**—Applies to network connections. This can include a PPP, SLIP, or ARAP connection.
- **Reverse Access**—Applies to reverse Telnet sessions.

When you create a named method list, you are defining a particular list of authorization methods for the indicated authorization type.

Once defined, method lists must be applied to specific lines or interfaces before any of the defined methods will be performed. The only exception is the default method list (which is named "default"). If the **aaa authorization** command for a particular authorization type is issued without a named method list specified, the default method list is automatically applied to all interfaces or lines except those that have a named method list explicitly defined. (A defined method list overrides the default method list.) If no default method list is defined, then no authorization takes place.

AAA Authorization Methods

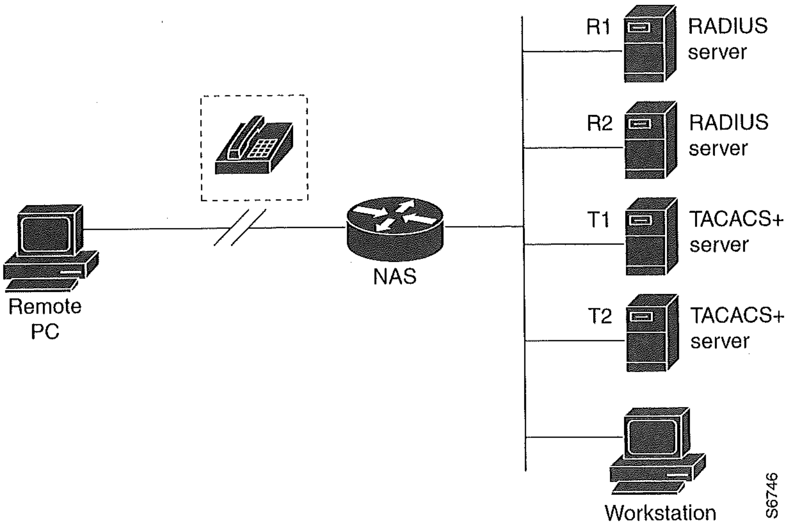
AAA supports five different methods of authorization:

- **TACACS+**—The network access server exchanges authorization information with the TACACS+ security daemon. TACACS+ authorization defines specific rights for users by associating attribute-value pairs, which are stored in a database on the TACACS+ security server, with the appropriate user.
- **If-Authenticated**—The user is allowed to access the requested function provided the user has been authenticated successfully.
- **None**—The network access server does not request authorization information; authorization is not performed over this line/interface.
- **Local**—The router or access server consults its local database, as defined by the **username** command, to authorize specific rights for users. Only a limited set of functions can be controlled via the local database.
- **RADIUS**—The network access server requests authorization information from the RADIUS security server. RADIUS authorization defines specific rights for users by associating attributes, which are stored in a database on the RADIUS server, with the appropriate user.

Method Lists and Server Groups

A server group is a way to group existing RADIUS or TACACS+ server hosts for use in method lists. Figure 6 shows a typical AAA network configuration that includes four security servers: R1 and R2 are RADIUS servers, and T1 and T2 are TACACS+ servers. R1 and R2 comprise the group of RADIUS servers. T1 and T2 comprise the group of TACACS+ servers.

Figure 6 Typical AAA Network Configuration



Using server groups, you can specify a subset of the configured server hosts and use them for a particular service. For example, server groups allow you to define R1 and R2 as separate server groups, and T1 and T2 as separate server groups. This means you can specify either R1 and T1 in the method list or R2 and T2 in the method list, which provides more flexibility in the way that you assign RADIUS and TACACS+ resources.

Server groups also can include multiple host entries for the same server, as long as each entry has a unique identifier. The combination of an IP address and a UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, authorization—the second host entry configured acts as fail-over backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server will try the second host entry configured on the same device for accounting services. (The RADIUS host entries will be tried in the order they are configured.)

For more information about configuring server groups and about configuring server groups based on DNIS numbers, refer to the "Configuring RADIUS" or "Configuring TACACS+" chapter.

AAA Authorization Types

Cisco IOS software supports five different types of authorization:

- **Auth-proxy**—Applies specific security policies on a per-user basis. For detailed information on the authentication proxy feature, refer to the "Configuring Authentication Proxy" chapter in the "Traffic Filtering and Firewalls" section of this book.
- **Commands**—Applies to the EXEC mode commands a user issues. Command authorization attempts authorization for all EXEC mode commands, including global configuration commands, associated with a specific privilege level.
- **EXEC**—Applies to the attributes associated with a user EXEC terminal session.
- **Network**—Applies to network connections. This can include a PPP, SLIP, or ARAP connection.
- **Reverse Access**—Applies to reverse Telnet sessions.

AAA Authorization Prerequisites

Before configuring authorization using named method lists, you must first perform the following tasks:

- Enable AAA on your network access server. For more information about enabling AAA on your Cisco router or access server, refer to the "AAA Overview" chapter.
- Configure AAA authentication. Authorization generally takes place after authentication and relies on authentication to work properly. For more information about AAA authentication, refer to the "Configuring Authentication" chapter.
- Define the characteristics of your RADIUS or TACACS+ security server if you are issuing RADIUS or TACACS+ authorization. For more information about configuring your Cisco network access server to communicate with your RADIUS security server, refer to the "Configuring RADIUS" chapter. For more information about configuring your Cisco network access server to communicate with your TACACS+ security server, refer to the "Configuring TACACS+" chapter.
- Define the rights associated with specific users by using the **username** command if you are issuing local authorization. For more information about the **username** command, refer to the *Cisco IOS Security Command Reference*.

AAA Authorization Configuration Task List

This section describes the following configuration tasks:

- Configuring AAA Authorization Using Named Method Lists
- Disabling Authorization for Global Configuration Commands
- Configuring Authorization for Reverse Telnet

For authorization configuration examples using the commands in this chapter, refer to the "Authorization Configuration Examples" section at the end of the this chapter.

Configuring AAA Authorization Using Named Method Lists

To configure AAA authorization using named method lists, use the following commands beginning in global configuration mode:

Step	Command	Purpose
1	**aaa authorization** {**auth-proxy** | **network** | **exec** | **commands** *level* | **reverse-access**} {**default** | *list-name*} [*method1* [*method2...*]]	Creates an authorization method list for a particular authorization type and enable authorization.
2	**line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*] **interface** *interface-type interface-number*	Enters the line configuration mode for the lines to which you want to apply the authorization method list. Alternately, enters the interface configuration mode for the interfaces to which you want to apply the authorization method list.

3	**authorization** {**arap** | **commands** *level* | **exec** | **reverse-access**} {**default** | *list-name*}
	ppp authorization {**default** | *list-name*}
		Applies the authorization list to a line or set of lines.
		Alternately, applies the authorization list to an interface or set of interfaces.

This section includes the following sections:

- Authorization Types
- Authorization Methods

Authorization Types

Named authorization method lists are specific to the indicated type of authorization.

To create a method list to enable authorization that applies specific security policies on a per-user basis, use the auth-proxy keyword. For detailed information on the authentication proxy feature, refer to the "Configuring Authentication Proxy" chapter in the "Traffic Filtering and Firewalls" section of this book.

To create a method list to enable authorization for all network-related service requests (including SLIP, PPP, PPP NCPs, and ARA protocols), use the **network** keyword.

To create a method list to enable authorization to determine if a user is allowed to run an EXEC shell, use the **exec** keyword.

To create a method list to enable authorization for specific, individual EXEC commands associated with a specific privilege level, use the **commands** keyword. (This allows you to authorize all commands associated with a specified command level from 0 to 15.)

To create a method list to enable authorization for reverse Telnet functions, use the **reverse-access** keyword.

For information about the types of authorization supported by the Cisco IOS software, refer to the "AAA Authorization Types" section.

Authorization Methods

To have the network access server request authorization information via a TACACS+ security server, use the **aaa authorization** command with the **group tacacs+** *method* keyword. For more specific information about configuring authorization using a TACACS+ security server, refer to the

"Configuring TACACS+" chapter. For an example of how to enable a TACACS+ server to authorize the use of network services, including PPP and ARA, see the "TACACS+ Authorization Examples" section at the end of this chapter.

To allow users to have access to the functions they request as long as they have been authenticated, use the **aaa authorization** command with the **if-authenticated** *method* keyword. If you select this method, all requested functions are automatically granted to authenticated users.

There may be times when you do not want to run authorization from a particular interface or line. To stop authorization activities on designated lines or interfaces, use the **none** *method* keyword. If you select this method, authorization is disabled for all actions.

To select local authorization, which means that the router or access server consults its local user database to determine the functions a user is permitted to use, use the **aaa authorization** command with the **local** *method* keyword. The functions associated with local authorization are defined by using the **username** global configuration command. For a list of permitted functions, refer to the "Configuring Authentication" chapter.

To have the network access server request authorization via a RADIUS security server, use the **radius** *method* keyword. For more specific information about configuring authorization using a RADIUS security server, refer to the "Configuring RADIUS" chapter.

To have the network access server request authorization via a RADIUS security server, use the **aaa authorization** command with the **group radius** *method* keyword. For more specific information about configuring authorization using a RADIUS security server, refer to the "Configuring RADIUS" chapter. For an example of how to enable a RADIUS server to authorize services, see the "RADIUS Authorization Example" section at the end of this chapter.

Note Authorization method lists for SLIP follow whatever is configured for PPP on the relevant interface. If no lists are defined and applied to a particular interface (or no PPP settings are configured), the default setting for authorization applies.

Disabling Authorization for Global Configuration Commands

The **aaa authorization** command with the keyword **commands** attempts authorization for all EXEC mode commands, including global configuration commands, associated with a specific privilege level. Because there are configuration commands that are identical to some EXEC-level commands, there can be some confusion in the authorization process. Using **no aaa authorization config-commands** stops the network access server from attempting configuration command authorization.

To disable AAA authorization for all global configuration commands, use the following command in global configuration mode:

Command	Purpose
`no aaa authorization config-commands`	Disables authorization for all global configuration commands.

Configuring Authorization for Reverse Telnet

Telnet is a standard terminal emulation protocol used for remote terminal connection. Normally, you log in to a network access server (typically through a dialup connection) and then use Telnet to access other network devices from that network access server. There are times, however, when it is necessary to establish a reverse Telnet session. In reverse Telnet sessions, the Telnet connection is established in the opposite direction—from inside a network to a network access server on the network periphery to gain access to modems or other devices connected to that network access server. Reverse Telnet is used to provide users with dialout capability by allowing them to Telnet to modem ports attached to a network access server.

It is important to control access to ports accessible through reverse Telnet. Failure to do so could, for example, allow unauthorized users free access to modems where they can trap and divert incoming calls or make outgoing calls to unauthorized destinations.

Authentication during reverse Telnet is performed through the standard AAA login procedure for Telnet. Typically the user has to provide a username and password to establish either a Telnet or reverse Telnet session. Reverse Telnet authorization provides an additional (optional) level of security by requiring authorization in addition to authentication. When enabled, reverse Telnet authorization can use RADIUS or TACACS+ to authorize whether or not this user is allowed reverse Telnet access to specific asynchronous ports, after the user successfully authenticates through the standard Telnet login procedure.

Reverse Telnet authorization offers the following benefits:

- An additional level of protection by ensuring that users engaged in reverse Telnet activities are indeed authorized to access a specific asynchronous port using reverse Telnet.
- An alternative method (other than access lists) to manage reverse Telnet authorization.

To configure a network access server to request authorization information from a TACACS+ or RADIUS server before allowing a user to establish a reverse Telnet session, use the following command in global configuration mode:

Command	Purpose
`aaa authorization reverse-access` *method1* [*method2* ...]	Configures the network access server to request authorization information before allowing a user to establish a reverse Telnet session.

This feature enables the network access server to request reverse Telnet authorization information from the security server, whether RADIUS or TACACS+. You must configure the specific reverse Telnet privileges for the user on the security server itself.

Authorization Attribute-Value Pairs

RADIUS and TACACS+ authorization both define specific rights for users by processing attributes, which are stored in a database on the security server. For both RADIUS and TACACS+, attributes are defined on the security server, associated with the user, and sent to the network access server where they are applied to the user's connection.

For a list of supported RADIUS attributes, refer to the "RADIUS Attributes" appendix. For a list of supported TACACS+ AV pairs, refer to the "TACACS+ Attribute-Value Pairs" appendix.

Authorization Configuration Examples

The following sections provide authorization configuration examples:

- Named Method List Configuration Example
- TACACS+ Authorization Examples
- RADIUS Authorization Example
- Reverse Telnet Authorization Examples

Named Method List Configuration Example

The following example shows how to configure a Cisco AS5200 (enabled for AAA and communication with a RADIUS security server) for AAA services to be provided by the RADIUS server. If the RADIUS server fails to respond, then the local database will be queried for authentication and authorization information, and accounting services will be handled by a TACACS+ server.

```
aaa new-model
aaa authentication login admins local
aaa authentication ppp dialins group radius local
aaa authorization network scoobee group radius local
aaa accounting network charley start-stop group radius

username root password ALongPassword

radius-server host alcatraz
radius-server key myRaDiUSpassWoRd

interface group-async 1
 group-range 1 16
 encapsulation ppp
 ppp authentication chap dialins
 ppp authorization scoobee
 ppp accounting charley

line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem dialin
```

The lines in this sample RADIUS AAA configuration are defined as follows:

- The **aaa new-model** command enables AAA network security services.
- The **aaa authentication login admins local** command defines a method list, admins, for login authentication.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list "dialins," which specifies that RADIUS authentication then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.
- The **aaa authorization network scoobee group radius local** command defines the network authorization method list named scoobee, which specifies that RADIUS authorization will be used on serial lines using PPP. If the RADIUS server fails to respond, then local network authorization will be performed.
- The **aaa accounting network charley start-stop group radius** command defines the network accounting method list named charley, which specifies that RADIUS accounting services (in this case, start and stop records for specific events) will be used on serial lines using PPP.
- The **username** command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.
- The **radius-server host** command defines the name of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **interface group-async** command selects and defines an asynchronous interface group.
- The **group-range** command defines the member asynchronous interfaces in the interface group.

- The **encapsulation ppp** command sets PPP as the encapsulation method used on the specified interfaces.
- The **ppp authentication chap dialins** command selects Challenge Handshake Authentication Protocol (CHAP) as the method of PPP authentication and applies the "dialins" method list to the specified interfaces.
- The **ppp authorization scoobee** command applies the scoobee network authorization method list to the specified interfaces.
- The **ppp accounting charley** command applies the charley network accounting method list to the specified interfaces.
- The **line** command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.
- The **autoselect ppp** command configures the Cisco IOS software to allow a PPP session to start up automatically on these selected lines.
- The **autoselect during-login** command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.
- The **login authentication admins** command applies the admins method list for login authentication.
- The **modem dialin** command configures modems attached to the selected lines to only accept incoming calls.

TACACS+ Authorization Examples

The following examples show how to use a TACACS+ server to authorize the use of network services, including PPP and ARA. If the TACACS+ server is not available or an error occurs during the authorization process, the fallback method (none) is to grant all authorization requests:

```
aaa authorization network default group tacacs+ none
```

The following example shows how to allow network authorization using TACACS+:

```
aaa authorization network default group tacacs+
```

The following example shows how to provide the same authorization, but it also creates address pools called *mci* and *att*:

```
aaa authorization network default group tacacs+
ip address-pool local
ip local-pool mci 172.16.0.1 172.16.0.255
ip local-pool att 172.17.0.1 172.17.0.255
```

These address pools can then be selected by the TACACS daemon. A sample configuration of the daemon follows:

```
user = mci_customer1 {
    login = cleartext "some password"
    service = ppp protocol = ip {
        addr-pool=mci
    }
}

user = att_customer1 {
    login = cleartext "some other password"
    service = ppp protocol = ip {
        addr-pool=att
    }
```

RADIUS Authorization Example

The following example shows how to configure the router to authorize using RADIUS:

```
aaa authorization exec default group radius if-authenticated
aaa authorization network default group radius
```

The lines in this sample RADIUS authorization configuration are defined as follows:

- The **aaa authorization exec default group radius if-authenticated** command configures the network access server to contact the RADIUS server to determine if users are permitted to start an EXEC shell when they log in. If an error occurs when the network access server contacts the RADIUS server, the fallback method is to permit the CLI to start, provided the user has been properly authenticated.

  The RADIUS information returned may be used to specify an autocommand or a connection access list be applied to this connection.

- The **aaa authorization network default group radius** command configures network authorization via RADIUS. This can be used to govern address assignment, the application of access lists, and various other per-user quantities.

Note Because no fallback method is specified in this example, authorization will fail if, for any reason, there is no response from the RADIUS server.

Reverse Telnet Authorization Examples

The following examples show how to cause the network access server to request authorization information from a TACACS+ security server before allowing a user to establish a reverse Telnet session:

```
aaa new-model
aaa authentication login default group tacacs+
aaa authorization reverse-access default group tacacs+
!
tacacs-server host 172.31.255.0
tacacs-server timeout 90
tacacs-server key goaway
```

The lines in this sample TACACS+ reverse Telnet authorization configuration are defined as follows:

- The **aaa new-model** command enables AAA.
- The **aaa authentication login default group tacacs+** command specifies TACACS+ as the default method for user authentication during login.
- The **aaa authorization reverse-access default group tacacs+** command specifies TACACS+ as the method for user authorization when trying to establish a reverse Telnet session.
- The **tacacs-server host** command identifies the TACACS+ server.
- The **tacacs-server timeout** command sets the interval of time that the network access server waits for the TACACS+ server to reply.
- The **tacacs-server key** command defines the encryption key used for all TACACS+ communications between the network access server and the TACACS+ daemon.

The following example shows how to configure a generic TACACS+ server to grant a user, pat, reverse Telnet access to port tty2 on the network access server named godzilla and to port tty5 on the network access server named gamera:

```
user = pat
  login = cleartext lab
  service = raccess {
    port#1 = godzilla/tty2
    port#2 = gamera/tty5
```

Note In this example, "godzilla" and "gamera" are the configured host names of network access servers, not DNS names or alias.

The following example shows how to configure the TACACS+ server (CiscoSecure) to grant a user named pat reverse Telnet access:

```
user = pat
profile_id = 90
profile_cycle = 1
member = Tacacs_Users
service=shell {
default cmd=permit
}
service=raccess {
allow "c2511e0" "tty1" ".*"
refuse ".*" ".*" ".*"
password = clear "goaway"
```

Note CiscoSecure only supports reverse Telnet using the command line interface in versions 2.1(*x*) through version 2.2(1).

An empty "service=raccess { }" clause permits a user to have unconditional access to network access server ports for reverse Telnet. If no "service=raccess" clause exists, the user is denied access to any port for reverse Telnet.

For more information about configuring TACACS+, refer to the "Configuring TACACS+" chapter. For more information about configuring CiscoSecure, refer to the *CiscoSecure Access Control Server User Guide*, version 2.1(2) or greater.

The following example shows how to cause the network access server to request authorization from a RADIUS security server before allowing a user to establish a reverse Telnet session:

```
aaa new-model
aaa authentication login default group radius
aaa authorization reverse-access default group radius
!
radius-server host 172.31.255.0
radius-server key go away
```

The lines in this sample RADIUS reverse Telnet authorization configuration are defined as follows:

- The **aaa new-model** command enables AAA.
- The **aaa authentication login default group radius** command specifies RADIUS as the default method for user authentication during login.
- The **aaa authorization reverse-access default group radius** command specifies RADIUS as the method for user authorization when trying to establish a reverse Telnet session.
- The **radius-server host** command identifies the RADIUS server.
- The **radius-server key** command defines the encryption key used for all RADIUS communications between the network access server and the RADIUS daemon.

The following example shows how to send a request to the RADIUS server to grant a user named "pat" reverse Telnet access at port tty2 on the network access server named godzilla:

```
Username = "pat"
Password = "goaway"
User-Service-Type = Shell-User
cisco-avpair = "raccess:port#1=godzilla/tty2"
```

An empty "raccess:port#1=nasname1/tty2" clause permits a user to have unconditional access to network access server ports for reverse Telnet. If no "raccess:port#1=nasname1/tty2" clause exists, the user is denied access to any port for reverse Telnet.

For more information about configuring RADIUS, refer to the "Configuring RADIUS" chapter.

Configuring Accounting

The AAA accounting feature enables you to track the services users are accessing as well as the amount of network resources they are consuming. When AAA accounting is enabled, the network access server reports user activity to the TACACS+ or RADIUS security server (depending on which security method you have implemented) in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server. This data can then be analyzed for network management, client billing, and/or auditing.

For a complete description of the accounting commands used in this chapter, refer to the "Accounting Commands" chapter in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

In This Chapter

This chapter has the following sections:

- Named Method Lists for Accounting
- AAA Accounting Types
- AAA Accounting Prerequisites
- AAA Accounting Configuration Task List
- Accounting Attribute-Value Pairs
- Accounting Configuration Example

Named Method Lists for Accounting

Like authentication and authorization method lists, method lists for accounting define the way accounting will be performed and the sequence in which these methods are performed.

Named accounting method lists enable you to designate a particular security protocol to be used on specific lines or interfaces for accounting services. The only exception is the default method list (which, by coincidence, is named "default"). The default method list is automatically applied to all interfaces except those that have a named method list explicitly defined. A defined method list overrides the default method list.

A method list is simply a named list describing the accounting methods to be queried (such as RADIUS or TACACS+), in sequence. Method lists enable you to designate one or more security protocols to be used for accounting, thus ensuring a backup system for accounting in case the initial method fails. Cisco IOS software uses the first method listed to support accounting; if that method fails to respond, the Cisco IOS software selects the next accounting method listed in the method list. This process continues until there is successful communication with a listed accounting method, or all methods defined are exhausted.

Note The Cisco IOS software attempts accounting with the next listed accounting method only when there is no response from the previous method. If accounting fails at any point in this cycle—meaning that the security server responds by denying the user access—the accounting process stops and no other accounting methods are attempted.

Accounting method lists are specific to the type of accounting being requested. AAA supports five different types of accounting:

- **Network**—Provides information for all PPP, SLIP, or ARAP sessions, including packet and byte counts.
- **EXEC**—Provides information about user EXEC terminal sessions of the network access server.
- **Commands**—Provides information about the EXEC mode commands that a user issues. Command accounting generates accounting records for all EXEC mode commands, including global configuration commands, associated with a specific privilege level.
- **Connection**—Provides information about all outbound connections made from the network access server, such as Telnet, local-area transport (LAT), TN3270, packet assembler/disassembler (PAD), and rlogin.
- **System**—Provides information about system-level events.

Note System accounting does not use named accounting lists; you can only define the default list for system accounting.

Once again, when you create a named method list, you are defining a particular list of accounting methods for the indicated accounting type.

Accounting method lists must be applied to specific lines or interfaces before any of the defined methods will be performed. The only exception is the default method list (which is named "default"). If the **aaa accounting** command for a particular accounting type is issued without a named method list specified, the default method list is automatically applied to all interfaces or lines except those that have a named method list explicitly defined. (A defined method list overrides the default method list.) If no default method list is defined, then no accounting takes place.

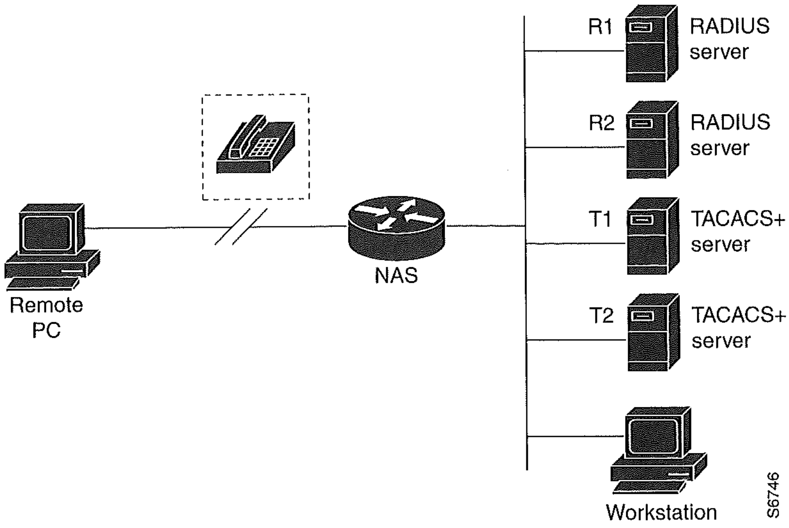
This section includes the following sections:

- Method Lists and Server Groups
- AAA Accounting Methods

Method Lists and Server Groups

A server group is a way to group existing RADIUS or TACACS+ server hosts for use in method lists. Figure 7 shows a typical AAA network configuration that includes four security servers: R1 and R2 are RADIUS servers, and T1 and T2 are TACACS+ servers. R1 and R2 comprise the group of RADIUS servers. T1 and T2 comprise the group of TACACS+ servers.

Figure 7 Typical AAA Network Configuration



Using server groups, you can specify a subset of the configured server hosts and use them for a particular service. For example, server groups allow you to define R1 and R2 as separate server groups, and T1 and T2 as separate server groups. This means you can specify either R1 and T1 in the method list or R2 and T2 in the method list, which provides more flexibility in the way that you assign RADIUS and TACACS+ resources.

Server groups also can include multiple host entries for the same server, as long as each entry has a unique identifier. The combination of an IP address and a UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, accounting—the second host entry configured acts as fail-over backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server will try the second host entry configured on the same device for accounting services. (The RADIUS host entries will be tried in the order they are configured.)

For more information about configuring server groups and about configuring server groups based on DNIS numbers, refer to the "Configuring RADIUS" or "Configuring TACACS+" chapter.

AAA Accounting Methods

Cisco IOS supports the following two methods for accounting:

- TACACS+—The network access server reports user activity to the TACACS+ security server in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server.
- RADIUS—The network access server reports user activity to the RADIUS security server in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server.

AAA Accounting Types

AAA supports five different accounting types:

- Network Accounting
- Connection Accounting
- EXEC Accounting
- System Accounting
- Command Accounting

Network Accounting

Network accounting provides information for all PPP, SLIP, or ARAP sessions, including packet and byte counts.

The following example shows the information contained in a RADIUS network accounting record for a PPP user who comes in through an EXEC session:

```
Wed Jun 25 04:44:45 1999
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 5
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "562"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "0000000D"
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"

Wed Jun 25 04:45:00 1999
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 5
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "562"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = "0000000E"
        Framed-IP-Address = "10.1.1.2"
        Framed-Protocol = PPP
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"

Wed Jun 25 04:47:46 1999
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 5
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "562"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = "0000000E"
        Framed-IP-Address = "10.1.1.2"
        Framed-Protocol = PPP
        Acct-Input-Octets = 3075
        Acct-Output-Octets = 167
        Acct-Input-Packets = 39
        Acct-Output-Packets = 9
        Acct-Session-Time = 171
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"

Wed Jun 25 04:48:45 1999
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 5
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "408"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "0000000D"
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ network accounting record for a PPP user who first started an EXEC session:

```
Wed Jun 25 04:00:35 1999        172.16.25.15     fgeorge    tty4     562/4327528
starttask_id=28       service=shell
Wed Jun 25 04:00:46 1999        172.16.25.15     fgeorge    tty4 562/4327528
starttask_id=30       addr=10.1.1.1    service=ppp
Wed Jun 25 04:00:49 1999        172.16.25.15     fgeorge    tty4     408/4327528
update        task_id=30        addr=10.1.1.1    service=ppp      protocol=ip
addr=10.1.1.1
Wed Jun 25 04:01:31 1999        172.16.25.15     fgeorge    tty4     562/4327528
stoptask_id=30        addr=10.1.1.1    service=ppp      protocol=ip     addr=10.1.1.1
bytes_in=2844         bytes_out=1682   paks_in=36       paks_out=24     elapsed_time=51
Wed Jun 25 04:01:32 1999        172.16.25.15     fgeorge    tty4     562/4327528
stoptask_id=28        service=shell    elapsed_time=57
```

Note The precise format of accounting packets records may vary depending on your particular security server daemon.

The following example shows the information contained in a RADIUS network accounting record for a PPP user who comes in through autoselect:

```
Wed Jun 25 04:30:52 1999
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 3
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "562"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = "0000000B"
        Framed-Protocol = PPP
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"
```

```
Wed Jun 25 04:36:49 1999
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 3
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "562"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = "0000000B"
        Framed-Protocol = PPP
        Framed-IP-Address = "10.1.1.1"
        Acct-Input-Octets = 8630
        Acct-Output-Octets = 5722
        Acct-Input-Packets = 94
        Acct-Output-Packets = 64
        Acct-Session-Time = 357
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ network accounting record for a PPP user who comes in through autoselect:

```
Wed Jun 25 04:02:19 1999        172.16.25.15     fgeorge    Async5   562/4327528
starttask_id=35       service=ppp
Wed Jun 25 04:02:25 1999        172.16.25.15     fgeorge    Async5   562/4327528
update        task_id=35       service=ppp      protocol=ip      addr=10.1.1.2
Wed Jun 25 04:05:03 1999        172.16.25.15     fgeorge    Async5   562/4327528
stoptask_id=35        service=ppp      protocol=ip      addr=10.1.1.2    bytes_in=3366
bytes_out=2149        paks_in=42       paks_out=28      elapsed_time=164
```

Connection Accounting

Connection accounting provides information about all outbound connections made from the network access server, such as Telnet, local-area transport (LAT), TN3270, packet assembler/disassembler (PAD), and rlogin.

The following example shows the information contained in a RADIUS connection accounting record for an outbound Telnet connection:

```
Wed Jun 25 04:28:00 1999
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 2
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329477"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = "00000008"
        Login-Service = Telnet
        Login-IP-Host = "171.68.202.158"
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"

Wed Jun 25 04:28:39 1999
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 2
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329477"
```

```
            Acct-Status-Type = Stop
            Acct-Authentic = RADIUS
            Service-Type = Login
            Acct-Session-Id = "00000008"
            Login-Service = Telnet
            Login-IP-Host = "171.68.202.158"
            Acct-Input-Octets = 10774
            Acct-Output-Octets = 112
            Acct-Input-Packets = 91
            Acct-Output-Packets = 99
            Acct-Session-Time = 39
            Acct-Delay-Time = 0
            User-Id = "fgeorge"
            NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ connection accounting record for an outbound Telnet connection:

```
Wed Jun 25 03:47:43 1999        172.16.25.15    fgeorge    tty3     5622329430/4327528
start     task_id=10      service=connection      protocol=telnet addr=171.68.202.158
cmd=telnet fgeorge-sun
Wed Jun 25 03:48:38 1999        172.16.25.15    fgeorge    tty3     5622329430/4327528
stop      task_id=10      service=connection      protocol=telnet addr=171.68.202.158
cmd=telnet fgeorge-sun      bytes_in=4467   bytes_out=96    paks_in=61      paks_out=72
elapsed_time=55
```

The following example shows the information contained in a RADIUS connection accounting record for an outbound rlogin connection:

```
Wed Jun 25 04:29:48 1999
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 2
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329477"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = "0000000A"
        Login-Service = Rlogin
        Login-IP-Host = "171.68.202.158"
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"

Wed Jun 25 04:30:09 1999
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 2
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329477"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = "0000000A"
        Login-Service = Rlogin
        Login-IP-Host = "171.68.202.158"
        Acct-Input-Octets = 18686
        Acct-Output-Octets = 86
        Acct-Input-Packets = 90
        Acct-Output-Packets = 68
        Acct-Session-Time = 22
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ connection accounting record for an outbound rlogin connection:

```
Wed Jun 25 03:48:46 1999        172.16.25.15    fgeorge    tty3     5622329430/4327528
start     task_id=12      service=connection      protocol=rlogin addr=171.68.202.158
cmd=rlogin fgeorge-sun /user fgeorge
Wed Jun 25 03:51:37 1999        172.16.25.15    fgeorge    tty3     5622329430/4327528
stop      task_id=12      service=connection      protocol=rlogin addr=171.68.202.158
cmd=rlogin fgeorge-sun /user fgeorge bytes_in=659926 bytes_out=138   paks_in=2378
paks_
out=1251          elapsed_time=171
```

The following example shows the information contained in a TACACS+ connection accounting record for an outbound LAT connection:

```
Wed Jun 25 03:53:06 1999        172.16.25.15    fgeorge    tty3     5622329430/4327528
start     task_id=18      service=connection      protocol=lat     addr=VAX
cmd=lat VAX
Wed Jun 25 03:54:15 1999        172.16.25.15    fgeorge    tty3     5622329430/4327528
stop      task_id=18      service=connection      protocol=lat     addr=VAX
cmd=lat VAX  bytes_in=0       bytes_out=0      paks_in=0       paks_out=0
elapsed_time=6
```

EXEC Accounting

EXEC accounting provides information about user EXEC terminal sessions (user shells) on the network access server, including username, date, start and stop times, the access server IP address, and (for dial-in users) the telephone number the call originated from.

The following example shows the information contained in a RADIUS EXEC accounting record for a dial-in user:

```
Wed Jun 25 04:26:23 1999
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 1
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329483"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "00000006"
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"

Wed Jun 25 04:27:25 1999
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 1
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329483"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "00000006"
        Acct-Session-Time = 62
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ EXEC accounting record for a dial-in user:

```
Wed Jun 25 03:46:21 1999        172.16.25.15    fgeorge    tty3    5622329430/4327528
start    task_id=2        service=shell
Wed Jun 25 04:08:55 1999        172.16.25.15    fgeorge    tty3    5622329430/4327528
stop     task_id=2        service=shell    elapsed_time=1354
```

The following example shows the information contained in a RADIUS EXEC accounting record for a Telnet user:

```
Wed Jun 25 04:48:32 1999
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 26
        User-Name = "fgeorge"
        Caller-ID = "171.68.202.158"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "00000010"
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"

Wed Jun 25 04:48:46 1999
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 26
        User-Name = "fgeorge"
        Caller-ID = "171.68.202.158"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "00000010"
        Acct-Session-Time = 14
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ EXEC accounting record for a Telnet user:

```
Wed Jun 25 04:06:53 1999        172.16.25.15    fgeorge    tty26    171.68.202.158
starttask_id=41       service=shell
Wed Jun 25 04:07:02 1999        172.16.25.15    fgeorge    tty26    171.68.202.158
stoptask_id=41        service=shell    elapsed_time=9
```

System Accounting

System accounting provides information about all system-level events (for example, when the system reboots or when accounting is turned on or off).

The following accounting record shows a typical TACACS+ system accounting record server indicating that AAA accounting has been turned off:

```
Wed Jun 25 03:55:32 1999        172.16.25.15    unknown unknown unknown start
task_id=25    service=system    event=sys_acct    reason=reconfigure
```

Note The precise format of accounting packets records may vary depending on your particular TACACS+ daemon.

The following accounting record shows a TACACS+ system accounting record indicating that AAA accounting has been turned on:

```
Wed Jun 25 03:55:22 1999        172.16.25.15    unknown unknown unknown stop
task_id=23    service=system  event=sys_acct  reason=reconfigure
```

Note Cisco's implementation of RADIUS does not support system accounting.

Additional tasks for measuring system resources are covered in other chapters in the Cisco IOS software configuration guides. For example, IP accounting tasks are described in the "Configuring IP Services" chapter in the *Cisco IOS IP and IP Routing Configuration Guide.*

Command Accounting

Command accounting provides information about the EXEC shell commands for a specified privilege level that are being executed on a network access server. Each command accounting record includes a list of the commands executed for that privilege level, as well as the date and time each command was executed, and the user who executed it.

The following example shows the information contained in a TACACS+ command accounting record for privilege level 1:

```
Wed Jun 25 03:46:47 1999        172.16.25.15    fgeorge   tty3    5622329430/4327528
stop      task_id=3      service=shell   priv-lvl=1      cmd=show version <cr>
Wed Jun 25 03:46:58 1999        172.16.25.15    fgeorge   tty3    5622329430/4327528
stop      task_id=4      service=shell   priv-lvl=1      cmd=show interfaces Ethernet 0
<cr>
Wed Jun 25 03:47:03 1999        172.16.25.15    fgeorge   tty3    5622329430/4327528
stop      task_id=5      service=shell   priv-lvl=1      cmd=show ip route <cr>
```

The following example shows the information contained in a TACACS+ command accounting record for privilege level 15:

```
Wed Jun 25 03:47:17 1999        172.16.25.15    fgeorge   tty3    5622329430/4327528
stop      task_id=6      service=shell   priv-lvl=15     cmd=configure terminal <cr>
Wed Jun 25 03:47:21 1999        172.16.25.15    fgeorge   tty3    5622329430/4327528
stop      task_id=7      service=shell   priv-lvl=15     cmd=interface Serial 0 <cr>
Wed Jun 25 03:47:29 1999        172.16.25.15    fgeorge   tty3    5622329430/4327528
stop      task_id=8      service=shell   priv-lvl=15     cmd=ip address 1.1.1.1
255.255.255.0 <cr>
```

Note Cisco's implementation of RADIUS does not support command accounting.

AAA Accounting Prerequisites

Before configuring accounting using named method lists, you must first perform the following tasks:

- Enable AAA on your network access server. For more information about enabling AAA on your Cisco router or access server, refer to the "AAA Overview" chapter.
- Define the characteristics of your RADIUS or TACACS+ security server if you are issuing RADIUS or TACACS+ authorization. For more information about configuring your Cisco network access server to communicate with your RADIUS security server, refer to the "Configuring RADIUS" chapter. For more information about configuring your Cisco network access server to communicate with your TACACS+ security server, refer to the "Configuring TACACS+" chapter.

AAA Accounting Configuration Task List

This section describes the following configuration tasks:

- Configuring AAA Accounting Using Named Method Lists
- Suppressing Generation of Accounting Records for Null Username Sessions
- Generating Interim Accounting Records
- Suppressing Generation of Accounting Records for Failed Login or Session
- Specifying Accounting NETWORK-Stop Records Before EXEC-Stop Records
- Monitoring Accounting

For accounting configuration examples using the commands in this chapter, refer to the "Accounting Configuration Example" section at the end of the this chapter.

Configuring AAA Accounting Using Named Method Lists

To configure AAA accounting using named method lists, use the following commands beginning in global configuration mode:

Step	Command	Purpose
1	**aaa accounting** {**system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} {**start-stop** | **stop-only** | **none**} [*method1* [*method2...*]]	Creates an accounting method list and enables accounting. The keyword *list-name* is a character string used to name the list you are creating.
2	**line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*] or **interface** *interface-type interface-number*	Enters the line configuration mode for the lines to which you want to apply the accounting method list. or Enters the interface configuration mode for the interfaces to which you want to apply the accounting method list.

3	**accounting** {**arap** | **commands** *level* | | **connection** | **exec** } {**default** | *list-name*}
	or
	ppp accounting {**default** | *list-name*}
		Applies the accounting method list to a line or set of lines.
		or
		Applies the accounting method list to an interface or set of interfaces.

Note System accounting does not use named method lists. For system accounting, you can only define the default method list.

This section includes the following sections:

- Accounting Types
- Accounting Record Types
- Accounting Methods

Accounting Types

Named accounting method lists are specific to the indicated type of accounting.

- **network**—To create a method list to enable authorization for all network-related service requests (including SLIP, PPP, PPP NCPs, and ARA protocols), use the **network** keyword. For example, to create a method list that provides accounting information for ARAP (network) sessions, use the **arap** keyword.
- **exec**—To create a method list that provides accounting records about user EXEC terminal sessions on the network access server, including username, date, start and stop times, use the **exec** keyword.
- **commands**—To create a method list that provides accounting information about specific, individual EXEC commands associated with a specific privilege level, use the **commands** keyword.
- **connection**—To create a method list that provides accounting information about all outbound connections made from the network access server, use the **connection** keyword.

Note System accounting does not support named method lists.

Accounting Record Types

For minimal accounting, use the **stop-only** keyword, which instructs the specified method (RADIUS or TACACS+) to send a stop record accounting notice at the end of the requested user process. For more accounting information, use the **start-stop** keyword to send a start accounting notice at the beginning of the requested event and a stop accounting notice at the end of the event. To stop all accounting activities on this line or interface, use the **none** keyword.

Accounting Methods

Table 10 lists the supported accounting methods.

Table 10 AAA Accounting Methods

Keyword	Description
group radius	Uses the list of all RADIUS servers for accounting.
group tacacs+	Uses the list of all TACACS+ servers for accounting.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for accounting as defined by the server group *group-name*.

The method argument refers to the actual method the authentication algorithm tries. Additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all other methods return an error, specify additional methods in the command. For example, to create a method list named acct_tac1 that specifies RADIUS as the backup method of authentication in the event that TACACS+ authentication returns an error, enter the following command:

```
aaa accounting network acct_tac1 stop-only group tacacs+ group radius
```

To create a default list that is used when a named list is *not* specified in the **aaa accounting** command, use the **default** keyword followed by the methods you want used in default situations. The default method list is automatically applied to all interfaces.

For example, to specify RADIUS as the default method for user authentication during login, enter the following command:

```
aaa accounting network default stop-only group radius
```

AAA accounting supports the following methods:

- **group tacacs**—To have the network access server send accounting information to a TACACS+ security server, use the **group tacacs+** *method* keyword. For more specific information about configuring TACACS+ for accounting services, refer to the "Configuring TACACS+" chapter.
- **group radius**—To have the network access server send accounting information to a RADIUS security server, use the **group radius** *method* keyword. For more specific information about configuring RADIUS for accounting services, refer to the "Configuring RADIUS" chapter.

Note Accounting method lists for SLIP follow whatever is configured for PPP on the relevant interface. If no lists are defined and applied to a particular interface (or no PPP settings are configured), the default setting for accounting applies.

- **group** *group-name*—To specify a subset of RADIUS or TACACS+ servers to use as the accounting method, use the **aaa accounting** command with the **group** *group-name* method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group loginrad**:

  ```
  aaa group server radius loginrad
   server 172.16.2.3
   server 172.16.2 17
   server 172.16.2.32
  ```

  This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *loginrad*.

  To specify **group loginrad** as the method of network accounting when no other method list has been defined, enter the following command:

  ```
  aaa accounting network default start-stop group loginrad
  ```

Before you can use a group name as the accounting method, you need to enable communication with the RADIUS or TACACS+ security server. For more information about establishing communication with a RADIUS server, refer to the "Configuring RADIUS" chapter. For more information about establishing communication with a TACACS+ server, refer to the "Configuring TACACS+" chapter.

Suppressing Generation of Accounting Records for Null Username Sessions

When AAA accounting is activated, the Cisco IOS software issues accounting records for all users on the system, including users whose username string, because of protocol translation, is NULL. An example of this is users who come in on lines where the **aaa authentication login** *method-list* **none** command is applied. To prevent accounting records from being generated for sessions that do not have usernames associated with them, use the following command in global configuration mode:

Command	Purpose
`aaa accounting suppress null-username`	Prevents accounting records from being generated for users whose username string is NULL.

Generating Interim Accounting Records

To enable periodic interim accounting records to be sent to the accounting server, use the following command in global configuration mode:

Command	Purpose
`aaa accounting update [newinfo] [periodic] number}`	Enables periodic interim accounting records to be sent to the accounting server.

When the **aaa accounting update** command is activated, the Cisco IOS software issues interim accounting records for all users on the system. If the keyword **newinfo** is used, interim accounting records will be sent to the accounting server every time there is new accounting information to report. An example of this would be when IPCP completes IP address negotiation with the remote peer. The interim accounting record will include the negotiated IP address used by the remote peer.

When used with the keyword **periodic**, interim accounting records are sent periodically as defined by the argument number. The interim accounting record contains all of the accounting information recorded for that user up to the time the interim accounting record is sent.

Caution Using the **aaa accounting update periodic** command can cause heavy congestion when many users are logged in to the network.

Suppressing Generation of Accounting Records for Failed Login or Session

When AAA accounting is activated, the Cisco IOS software does not generate accounting records for system users who fail login authentication, or who succeed in login authentication but fail PPP negotiation for some reason.

To specify that accounting stop records be generated for users who fail to authenticate at login or during session negotiation, use the following command in global configuration mode:

Command	Purpose
`aaa accounting send stop-record authentication failure`	Generates stop-records for users who fail to authenticate at login or during session negotiation using PPP.

Specifying Accounting NETWORK-Stop Records Before EXEC-Stop Records

For PPP users who start EXEC terminal sessions, you can specify that NETWORK records be generated before EXEC-stop records. In some cases, such as billing customers for specific services, is can be desirable to keep network start and stop records together, essentially "nesting" them within the framework of the EXEC start and stop messages. For example, a user dialing in using PPP can create the following records: EXEC-start, NETWORK-start, EXEC-stop, NETWORK-stop. By nesting the accounting records, NETWORK-stop records follow NETWORK-start messages: EXEC-start, NETWORK-start, NETWORK-stop, EXEC-stop.

To nest accounting records for user sessions, use the following command in global configuration mode:

Command	Purpose
`aaa accounting nested`	Nests network accounting records.

Monitoring Accounting

No specific **show** command exists for either RADIUS or TACACS+ accounting. To obtain accounting records displaying information about users currently logged in, use the following command in privileged EXEC mode:

Command	Purpose
`show accounting`	Steps through all active sessions and print all the accounting records for the actively accounted functions.

Accounting Attribute-Value Pairs

The network access server monitors the accounting functions defined in either TACACS+ attribute-value (AV) pairs or RADIUS attributes, depending on which security method you have implemented. For a list of supported RADIUS accounting attributes, refer to the "RADIUS Attributes" appendix. For a list of supported TACACS+ accounting AV pairs, refer to the "TACACS+ AV Pairs" appendix.

Accounting Configuration Example

This section contains the Named Method List Configuration Example.

Named Method List Configuration Example

The following example shows how to configure a Cisco AS5200 (enabled for AAA and communication with a RADIUS security server) for AAA services to be provided by the RADIUS server. If the RADIUS server fails to respond, then the local database will be queried for authentication and authorization information, and accounting services will be handled by a TACACS+ server.

```
aaa new-model
aaa authentication login admins local
aaa authentication ppp dialins goup radius local
aaa authorization network scoobee group radius local
aaa accounting network charley start-stop group radius group tacacs+

username root password ALongPassword

tacacs-server host 172.31.255.0
tacacs-server key goaway

radius-server host 172.16.2.7
radius-server key myRaDiUSpassWoRd
```

```
interface group-async 1
 group-range 1 16
 encapsulation ppp
 ppp authentication chap dialins
 ppp authorization scoobee
 ppp accounting charley

line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem dialin
```

The lines in this sample RADIUS AAA configuration are defined as follows:

- The **aaa new-model** command enables AAA network security services.
- The **aaa authentication login admins local** command defines a method list, admins, for login authentication.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list "dialins," which specifies that RADIUS authentication then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.
- The **aaa authorization network scoobee group radius local** command defines the network authorization method list named scoobee, which specifies that RADIUS authorization will be used on serial lines using PPP. If the RADIUS server fails to respond, then local network authorization will be performed.
- The **aaa accounting network charley start-stop group radius group tacacs+** command defines the network accounting method list named charley, which specifies that RADIUS accounting services (in this case, start and stop records for specific events) will be used on serial lines using PPP. If the RADIUS server fails to respond, accounting services will be handled by a TACACS+ server.
- The **username** command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.
- The **tacacs-server host** command defines the name of the TACACS+ server host.
- The **tacacs-server key** command defines the shared secret text string between the network access server and the TACACS+ server host.
- The **radius-server host** command defines the name of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **interface group-async** command selects and defines an asynchronous interface group.
- The **group-range** command defines the member asynchronous interfaces in the interface group.
- The **encapsulation ppp** command sets PPP as the encapsulation method used on the specified interfaces.
- The **ppp authentication chap dialins** command selects Challenge Handshake Authentication Protocol (CHAP) as the method of PPP authentication and applies the "dialins" method list to the specified interfaces.
- The **ppp authorization scoobee** command applies the scoobee network authorization method list to the specified interfaces.

- The **ppp accounting charley** command applies the charley network accounting method list to the specified interfaces.
- The **line** command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.
- The **autoselect ppp** command configures the Cisco IOS software to allow a PPP session to start up automatically on these selected lines.
- The **autoselect during-login** command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.
- The **login authentication admins** command applies the admins method list for login authentication.
- The **modem dialin** command configures modems attached to the selected lines to only accept incoming calls.

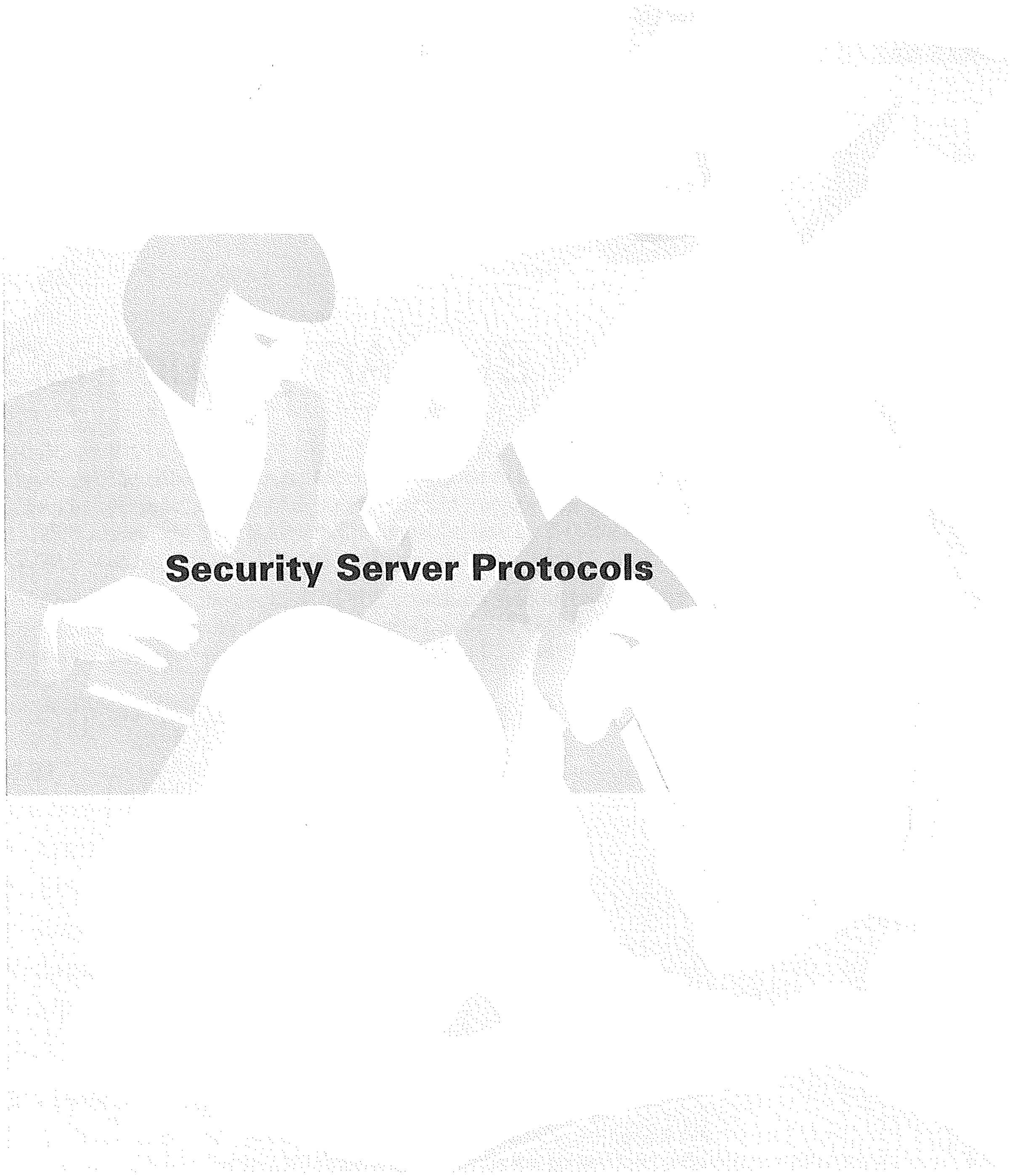
The **show accounting** command yields the following output for the preceding configuration:

```
Active Accounted actions on tty1, User rubble Priv 1
 Task ID 5, Network Accounting record, 00:00:52 Elapsed
 task_id=5 service=ppp protocol=ip address=10.0.0.98
```

Table 11 describes the fields contained in the preceding output.

Table 11 show accounting Field Descriptions

Field	Description
Active Accounted actions on	Terminal line or interface name user with which the user logged in.
User	User's ID.
Priv	User's privilege level.
Task ID	Unique identifier for each accounting session.
Accounting Record	Type of accounting session.
Elapsed	Length of time (hh:mm:ss) for this session type.
attribute=value	AV pairs associated with this accounting session.



Security Server Protocols

Configuring RADIUS

This chapter describes the Remote Authentication Dial-In User Service (RADIUS) security system, defines its operation, and identifies appropriate and inappropriate network environments for using RADIUS technology. The "RADIUS Configuration Task List" section describes how to configure RADIUS with the authentication, authorization, and accounting (AAA) command set.

For a complete description of the RADIUS commands used in this chapter, refer to the "RADIUS Commands" chapter in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

In This Chapter

This chapter includes the following sections:

- RADIUS Overview
- RADIUS Operation
- RADIUS Configuration Task List
- RADIUS Attributes
- RADIUS Configuration Examples

RADIUS Overview

RADIUS is a distributed client/server system that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

RADIUS is a fully open protocol, distributed in source code format, that can be modified to work with any security system currently available on the market.

Cisco supports RADIUS under its AAA security paradigm. RADIUS can be used with other AAA security protocols, such as TACACS+, Kerberos, or local username lookup. RADIUS is supported on all Cisco platforms.

RADIUS has been implemented in a variety of network environments that require high levels of security while maintaining network access for remote users.

Use RADIUS in the following network environments that require access security:

- Networks with multiple-vendor access servers, each supporting RADIUS. For example, access servers from several vendors use a single RADIUS server-based security database. In an IP-based network with multiple vendors' access servers, dial-in users are authenticated through a RADIUS server that has been customized to work with the Kerberos security system.
- Turnkey network security environments in which applications support the RADIUS protocol, such as in an access environment that uses a "smart card" access control system. In one case, RADIUS has been used with Enigma's security cards to validate users and grant access to network resources.
- Networks already using RADIUS. You can add a Cisco router with RADIUS to the network. This might be the first step when you make a transition to a Terminal Access Controller Access Control System Plus (TACACS+) server.
- Networks in which a user must only access a single service. Using RADIUS, you can control user access to a single host, to a single utility such as Telnet, or to a single protocol such as Point-to-Point Protocol (PPP). For example, when a user logs in, RADIUS identifies this user as having authorization to run PPP using IP address 10.2.3.4 and the defined access list is started.
- Networks that require resource accounting. You can use RADIUS accounting independent of RADIUS authentication or authorization. The RADIUS accounting functions allow data to be sent at the start and end of services, indicating the amount of resources (such as time, packets, bytes, and so on) used during the session. An Internet service provider (ISP) might use a freeware-based version of RADIUS access control and accounting software to meet special security and billing needs.

RADIUS is not suitable in the following network security situations:

- Multiprotocol access environments. RADIUS does not support the following protocols:
    - AppleTalk Remote Access (ARA)
    - NetBIOS Frame Control Protocol (NBFCP)
    - NetWare Asynchronous Services Interface (NASI)
    - X.25 PAD connections
- Router-to-router situations. RADIUS does not provide two-way authentication. RADIUS can be used to authenticate from one router to a non-Cisco router if the non-Cisco router requires RADIUS authentication.
- Networks using a variety of services. RADIUS generally binds a user to one service model.

RADIUS Operation

When a user attempts to log in and authenticate to an access server using RADIUS, the following steps occur:

1. The user is prompted for and enters a username and password.
2. The username and encrypted password are sent over the network to the RADIUS server.
3. The user receives one of the following responses from the RADIUS server:
    (a) ACCEPT—The user is authenticated.
    (b) REJECT—The user is not authenticated and is prompted to reenter the username and password, or access is denied.

(c) CHALLENGE—A challenge is issued by the RADIUS server. The challenge collects additional data from the user.

(d) CHANGE PASSWORD—A request is issued by the RADIUS server, asking the user to select a new password.

The ACCEPT or REJECT response is bundled with additional data that is used for EXEC or network authorization. You must first complete RADIUS authentication before using RADIUS authorization. The additional data included with the ACCEPT or REJECT packets consists of the following:

- Services that the user can access, including Telnet, rlogin, or local-area transport (LAT) connections, and PPP, Serial Line Internet Protocol (SLIP), or EXEC services.
- Connection parameters, including the host or client IP address, access list, and user timeouts.

RADIUS Configuration Task List

To configure RADIUS on your Cisco router or access server, you must perform the following tasks:

- Use the **aaa new-model** global configuration command to enable AAA. AAA must be configured if you plan to use RADIUS. For more information about using the **aaa new-model** command, refer to the "AAA Overview" chapter.
- Use the **aaa authentication** global configuration command to define method lists for RADIUS authentication. For more information about using the **aaa authentication** command, refer to the "Configuring Authentication" chapter.
- Use **line** and **interface** commands to enable the defined method lists to be used. For more information, refer to the "Configuring Authentication" chapter.

The following configuration tasks are optional:

- If needed, use the **aaa server group** command to group selected RADIUS hosts for specific services. For more information about using the **aaa server group** command, refer to the "Configuring AAA Server Groups" section in this chapter.
- If needed, use the **aaa dnis map** command to select RADIUS server groups based on DNIS number. To use this command, you must define RADIUS server groups using the **aaa server group** command. For more information about using the **aaa dnis map** command, refer to the "Configuring AAA Server Group Selection Based on DNIS" section in this chapter.
- If needed, use the **aaa authorization** global command to authorize specific user functions. For more information about using the **aaa authorization** command, refer to the "Configuring Authorization" chapter.
- If needed, use the **aaa accounting** command to enable accounting for RADIUS connections. For more information about using the **aaa accounting** command, refer to the "Configuring Accounting" chapter.

This section describes how to set up RADIUS for authentication, authorization, and accounting on your network, and includes the following sections:

- Configuring Router to RADIUS Server Communication (Required)
- Configuring Router to Use Vendor-Specific RADIUS Attributes (Optional)
- Configuring Router for Vendor-Proprietary RADIUS Server Communication (Optional)
- Configuring Router to Query RADIUS Server for Static Routes and IP Addresses (Optional)
- Configuring Router to Expand Network Access Server Port Information (Optional)
- Configuring AAA Server Groups (Optional)

- Configuring AAA Server Group Selection Based on DNIS (Optional)
- Specifying RADIUS Authentication
- Specifying RADIUS Authorization (Optional)
- Specifying RADIUS Accounting (Optional)

For RADIUS configuration examples using the commands in this chapter, refer to the "RADIUS Configuration Examples" section at the end of the this chapter.

Configuring Router to RADIUS Server Communication

The RADIUS host is normally a multiuser system running RADIUS server software from Livingston, Merit, Microsoft, or another software provider. Configuring router to RADIUS server communication can have several components:

- Host name or IP address
- Authentication destination port
- Accounting destination port
- Timeout period
- Retransmission value
- Key string

RADIUS security servers are identified on the basis of their host name or IP address, host name and specific UDP port numbers, or IP address and specific UDP port numbers. The combination of the IP address and UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, accounting—the second host entry configured acts as fail-over backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server will try the second host entry configured on the same device for accounting services. (The RADIUS host entries will be tried in the order they are configured.)

A RADIUS server and a Cisco router use a shared secret text string to encrypt passwords and exchange responses.To configure RADIUS to use the AAA security commands, you must specify the host running the RADIUS server daemon and a secret text (key) string that it shares with the router.

The timeout, retransmission, and encryption key values are configurable globally for all RADIUS servers, on a per-server basis, or in some combination of global and per-server settings. To apply these settings globally to all RADIUS servers communicating with the router, use the three unique global commands: **radius-server timeout**, **radius-server retransmit**, and **radius-server key**. To apply these values on a specific RADIUS server, use the **radius-server host** command.

Note You can configure both global and per-server timeout, retransmission, and key value commands simultaneously on the same Cisco network access server. If both global and per-server functions are configured on a router, the per-server timer, retransmission, and key value commands override global timer, retransmission, and key value commands.

To configure per-server RADIUS server communication, use the following command in global configuration mode:

Command	Purpose
radius-server host {*hostname* | *ip-address*} [**auth-port** *port-number*] [**acct-port** *port-number*] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*]	Specifies the IP address or host name of the remote RADIUS server host and assign authentication and accounting destination port numbers. Use the **auth-port** *port-number* option to configure a specific UDP port on this RADIUS server to be used solely for authentication. Use the **acct-port** *port-number* option to configure a specific UDP port on this RADIUS server to be used solely for accounting. To configure the network access server to recognize more than one host entry associated with a single IP address, simply repeat this command as many times as necessary, making sure that each UDP port number is different. Set the timeout, retransmit, and encryption key values to use with the specific RADIUS host. If no timeout is set, the global value is used; otherwise, enter a value in the range 1 to 1000. If no retransmit value is set, the global value is used; otherwise enter a value in the range 1 to 1000. If no key string is specified, the global value is used. **Note** The key is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the **radius-server host** command syntax because the leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks themselves are part of the key.

To configure global communication settings between the router and a RADIUS server, use the following **radius-server** global configuration commands:

Step	Command	Purpose
1	**radius-server key** *string*	Specifies the shared secret text string used between the router and a RADIUS server.
2	**radius-server retransmit** *retries*	Specifies the number of times the router transmits each RADIUS request to the server before giving up (the default is three).
3	**radius-server timeout** *seconds*	Specifies the number of seconds a router waits for a reply to a RADIUS request before retransmitting the request.
4	**radius-server deadtime** *minutes*	Specifies the number of minutes a RADIUS server, which is not responding to authentication requests, is passed over by requests for RADIUS authentication.

Configuring Router to Use Vendor-Specific RADIUS Attributes

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific information between the network access server and the RADIUS server by using the vendor-specific attribute (attribute 26). Vendor-specific attributes (VSAs) allow vendors to support their own extended attributes not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. Cisco's vendor-ID is 9, and the supported option has vendor-type 1, which is named "cisco-avpair." The value is a string of the following format:

```
protocol : attribute sep value *
```

"Protocol" is a value of the Cisco "protocol" attribute for a particular type of authorization. "Attribute" and "value" are an appropriate attribute-value (AV) pair defined in the Cisco TACACS+ specification, and "sep" is "=" for mandatory attributes and "*" for optional attributes. This allows the full set of features available for TACACS+ authorization to also be used for RADIUS.

For example, the following AV pair causes Cisco's "multiple named ip address pools" feature to be activated during IP authorization (during PPP's IPCP address assignment):

```
cisco-avpair= "ip:addr-pool=first"
```

The following example shows how to cause a user logging in from a network access server to have immediate access to EXEC commands:

```
cisco-avpair= "shell:priv-lvl=15"
```

Other vendors have their own unique vendor-IDs, options, and associated VSAs. For more information about vendor-IDs and VSAs, refer to RFC 2138, "Remote Authentication Dial-In User Service (RADIUS)."

To configure the network access server to recognize and use VSAs, use the following command in global configuration mode:

Command	Purpose
`radius-server vsa send [accounting | authentication]`	Enables the network access server to recognize and use VSAs as defined by RADIUS IETF attribute 26.

For a complete list of RADIUS attributes or more information about vendor-specific attribute 26, refer to the "RADIUS Attributes" appendix.

Configuring Router for Vendor-Proprietary RADIUS Server Communication

Although an Internet Engineering Task Force (IETF) draft standard for RADIUS specifies a method for communicating vendor-proprietary information between the network access server and the RADIUS server, some vendors have extended the RADIUS attribute set in a unique way. Cisco IOS software supports a subset of vendor-proprietary RADIUS attributes.

As mentioned earlier, to configure RADIUS (whether vendor-proprietary or IETF draft-compliant), you must specify the host running the RADIUS server daemon and the secret text string it shares with the Cisco device. You specify the RADIUS host and secret text string by using the **radius-server** commands. To identify that the RADIUS server is using a vendor-proprietary implementation of RADIUS, use the **radius-server host non-standard** command. Vendor-proprietary attributes will not be supported unless you use the **radius-server host non-standard** command.

To specify a vendor-proprietary RADIUS server host and a shared secret text string, use the following commands in global configuration mode:

Step	Command	Purpose
1	`radius-server host {hostname	ip-address} non-standard`	Specifies the IP address or host name of the remote RADIUS server host and identify that it is using a vendor-proprietary implementation of RADIUS.
2	`radius-server key string`	Specifies the shared secret text string used between the router and the vendor-proprietary RADIUS server. The router and the RADIUS server use this text string to encrypt passwords and exchange responses.

Configuring Router to Query RADIUS Server for Static Routes and IP Addresses

Some vendor-proprietary implementations of RADIUS let the user define static routes and IP pool definitions on the RADIUS server instead of on each individual network access server in the network. Each network access server then queries the RADIUS server for static route and IP pool information.

To have the Cisco router or access server query the RADIUS server for static routes and IP pool definitions when the device first starts up, use the following command in global configuration mode:

Command	Purpose
`radius-server configure-nas`	Tells the Cisco router or access server to query the RADIUS server for the static routes and IP pool definitions used throughout its domain.

Note Because the **radius-server configure-nas** command is performed when the Cisco router starts up, it will not take effect until you issue a **copy system:running config nvram:startup-config** command.

Configuring Router to Expand Network Access Server Port Information

There are some situations when PPP or login authentication occurs on an interface different from the interface on which the call itself comes in. For example, in a V.120 ISDN call, login or PPP authentication occurs on a virtual asynchronous interface "ttt" but the call itself occurs on one of the channels of the ISDN interface.

The **radius-server attribute nas-port extended** command configures RADIUS to expand the size of the NAS-Port attribute (RADIUS IETF attribute 5) field to 32 bits. The upper 16 bits of the NAS-Port attribute display the type and number of the controlling interface; the lower 16 bits indicate the interface undergoing authentication.

To display expanded interface information in the NAS-Port attribute field, use the following command in global configuration mode:

Command	Purpose
`radius-server attribute nas-port extended`	Expands the size of the NAS-Port attribute from 16 to 32 bits to display extended interface information.

Note This command replaces the deprecated **radius-server extended-portnames** command.

On platforms with multiple interfaces (ports) per slot, the Cisco RADIUS implementation will not provide a unique NAS-Port attribute that permits distinguishing between the interfaces. For example, if a dual PRI interface is in slot 1, calls on both Serial1/0:1 and Serial1/1:1 will appear as NAS-Port = 20101.

Once again, this is because of the 16-bit field size limitation associated with RADIUS IETF NAS-Port attribute. In this case, the solution is to replace the NAS-Port attribute with a vendor-specific attribute (RADIUS IETF attribute 26). Cisco's vendor-ID is 9, and the Cisco-NAS-Port attribute is subtype 2. Vendor-specific attributes (VSAs) can be turned on by entering the **radius-server vsa send** command. The port information in this attribute is provided and configured using the **aaa nas port extended** command.

To replace the NAS-Port attribute with RADIUS IETF attribute 26 and to display extended field information, use the following commands in global configuration mode:

Step	Command	Purpose
1	`radius-server vsa send [accounting	authentication]`	Enables the network access server to recognize and use vendor-specific attributes as defined by RADIUS IETF attribute 26.
2	`aaa nas port extended`	Expands the size of the VSA NAS-Port field from 16 to 32 bits to display extended interface information.

The standard NAS-Port attribute (RADIUS IETF attribute 5) will continue to be sent. If you do not want this information to be sent, you can suppress it by using the **no radius-server attribute nas-port** command. When this command is configured, the standard NAS-Port attribute will no longer be sent.

For a complete list of RADIUS attributes, refer to the "RADIUS Attributes" appendix.

Configuring AAA Server Groups

Configuring the router to use AAA server groups provides a way to group existing server hosts. This allows you to select a subset of the configured server hosts and use them for a particular service. A server group is used in conjunction with a global server-host list. The server group lists the IP addresses of the selected server hosts.

Server groups also can include multiple host entries for the same server, as long as each entry has a unique identifier. The combination of an IP address and a UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, accounting—the second host entry configured acts as fail-over backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server will try the second host entry configured on the same device for accounting services. (The RADIUS host entries will be tried in the order they are configured.)

To define a server host with a server group name, enter the following commands in global configuration mode. The listed server must exist in global configuration mode:

Step	Command	Purpose
1	`Router(config)# radius-server host {hostname	ip-address} [auth-port port-number] [acct-port port-number] [timeout seconds] [retransmit retries] [key string]`	Specifies and defines the IP address of the server host before configuring the AAA server-group. Refer to the "Configuring Router to RADIUS Server Communication" section for more information on the **radius-server host** command.
2	`Router(config-if)# aaa group server {radius	tacacs+} group-name`	Defines the AAA server-group with a group name. All members of a group must be the same type; that is, RADIUS or TACACS+. This command puts the router in server group subconfiguration mode.
3	`Router(config-sg)# server ip-address [auth-port port number] [acct-port port-number]`	Associates a particular RADIUS server with the defined server group. Each security server is identified by its IP address and UDP port number. Repeat this step for each RADIUS server in the AAA server group. **Note** Each server in the group must be defined previously using the **radius-server host** command.

Configuring AAA Server Group Selection Based on DNIS

Cisco IOS allows you to authenticate users to a particular AAA server group based on the session's Dialed Number Identification Service (DNIS) number. Any phone line (a regular home phone or a commercial T1/PRI line) can be associated with several phone numbers. The DNIS number identifies the number that was called to reach you.

For example, suppose you want to share the same phone number with several customers, but you want to know which customer is calling before you pick up the phone. You can customize how you answer the phone because DNIS allows you to know which customer is calling when you answer.

Cisco routers with either ISDN or internal modems can receive the DNIS number. This functionality allows users to assign different RADIUS server groups for different customers (that is, different RADIUS servers for different DNIS numbers). Additionally, using server groups you can specify the same server group for AAA services or a separate server group for each AAA service.

Cisco IOS software provides the flexibility to implement authentication and accounting services in several ways:

- Globally—AAA services are defined using global configuration access list commands and applied in general to all interfaces on a specific network access server.
- Per Interface—AAA services are defined using interface configuration commands and applied specifically to the interface being configured on a specific network access server.
- DNIS mapping—You can use DNIS to specify an AAA server to supply AAA services.

Because each of these AAA configuration methods can be configured simultaneously, Cisco has established an order of precedence to determine which server or groups of servers provide AAA services. The order of precedence is as follows:

- Per DNIS—If you configure the network access server to use DNIS to identify/determine which server group provides AAA services, then this method takes precedence over any additional AAA selection method.

- Per interface—If you configure the network access server per interface to use access lists to determine how a server provides AAA services, this method takes precedence over any global configuration AAA access lists.
- Globally—If you configure the network access server by using global AAA access lists to determine how the security server provides AAA services, this method has the least precedence.

Note Prior to configuring AAA Server Group Selection Based on DNIS, you must configure the list of RADIUS server hosts and configure the AAA server groups. See the "Configuring Router to RADIUS Server Communication" and "Configuring AAA Server Groups" sections in this chapter.

To configure the router to select a particular AAA server group based on the DNIS of the server group, configure DNIS mapping. To map a server group with a group name with DNIS number, use the following commands in global configuration mode:

Step	Command	Purpose
1	`Router(config)#` **`aaa dnis map enable`**	Enables DNIS mapping.
2	`Router(config)#` **`aaa dnis map`** *`dnis-number`* **`authentication ppp group`** `server-`*`group-name`*	Maps a DNIS number to a defined AAA server group; the servers in this server group are being used for authentication.
3	`Router(config)#` **`aaa dnis map`** *`dnis-number`* **`accounting network`** `[`**`none`** `|` **`start-stop`** `|` **`stop-only`**`]` **`group`** *`server-group-name`*	Maps a DNIS number to a defined AAA server group; the servers in this server group are being used for accounting.

Specifying RADIUS Authentication

After you have identified the RADIUS server and defined the RADIUS authentication key, you need to define method lists for RADIUS authentication. Because RADIUS authentication is facilitated through AAA, you need to enter the **aaa authentication** command, specifying RADIUS as the authentication method. For more information, refer to the "Configuring Authentication" chapter.

Specifying RADIUS Authorization

AAA authorization lets you set parameters that restrict a user's network access. Authorization using RADIUS provides one method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, ARA, and Telnet. Because RADIUS authorization is facilitated through AAA, you need to issue the **aaa authorization** command, specifying RADIUS as the authorization method. For more information, refer to the "Configuring Authorization" chapter.

Specifying RADIUS Accounting

The AAA accounting feature enables you to track the services users are accessing as well as the amount of network resources they are consuming. Because RADIUS accounting is facilitated through AAA, you need to issue the **aaa accounting** command, specifying RADIUS as the accounting method. For more information, refer to the "Configuring Accounting" chapter.

RADIUS Attributes

The network access server monitors the RADIUS authorization and accounting functions defined by RADIUS attributes in each user-profile. For a list of supported RADIUS attributes, refer to the "RADIUS Attributes" appendix.

This section includes the following sections:

- Vendor-Proprietary RADIUS Attributes
- RADIUS Tunnel Attributes

Vendor-Proprietary RADIUS Attributes

An Internet Engineering Task Force (IETF) draft standard for RADIUS specifies a method for communicating vendor-proprietary information between the network access server and the RADIUS server. Some vendors, nevertheless, have extended the RADIUS attribute set in a unique way. Cisco IOS software supports a subset of vendor-proprietary RADIUS attributes. For a list of supported vendor-proprietary RADIUS attributes, refer to the "RADIUS Attributes" appendix.

RADIUS Tunnel Attributes

RADIUS is a security server authentication, authorization, and accounting (AAA) protocol originally developed by Livingston, Inc. RADIUS uses attribute value (AV) pairs to communicate information between the security server and the network access server. RFC 2138 and RFC 2139 describe the basic functionality of RADIUS and the original set of Internet Engineering Task Force (IETF)-standard AV pairs used to send AAA information. Two draft IETF standards, "RADIUS Attributes for Tunnel Protocol Support" and "RADIUS Accounting Modifications for Tunnel Protocol Support," extend the IETF-defined set of AV pairs to include attributes specific to virtual private dial-up networks (VPDNs); these attributes are used to carry the tunneling information between the RADIUS server and the tunnel initiator.

In the past, Cisco routers and access servers have only been able to support VPDN tunnel attributes by using extensions to the Cisco vendor-specific attribute 26. This feature enables Cisco routers and access servers to support the new RADIUS IETF-standard VPDN tunnel attributes. These new RADIUS IETF-standard attributes are listed in the "RADIUS Attributes" appendix. Refer to the following three configuration examples later in this chapter:

- L2TP Access Concentrator Examples
- L2TP Network Server Example
- RADIUS User Profile with RADIUS Tunneling Attributes Example

For more information about L2F, L2TP, VPN, or VPDN, refer to the *Cisco IOS Dial Services Configuration Guide: Network Services.*

RADIUS Configuration Examples

The following sections provide RADIUS configuration examples:

- RADIUS Authentication and Authorization Example
- RADIUS Authentication, Authorization, and Accounting Example
- Vendor-Proprietary RADIUS Configuration Example
- RADIUS Server with Server-Specific Values Example

- Multiple RADIUS Servers with Global and Server-Specific Values Example
- Multiple RADIUS Server Entries for the Same Server IP Address Example
- RADIUS Server Group Examples
- Multiple RADIUS Server Entries Using AAA Server Groups Example
- AAA Server Group Selection Based on DNIS Example
- L2TP Access Concentrator Examples
- L2TP Network Server Example
- RADIUS User Profile with RADIUS Tunneling Attributes Example

RADIUS Authentication and Authorization Example

The following example shows how to configure the router to authenticate and authorize using RADIUS:

```
aaa authentication login use-radius group radius local
aaa authentication ppp user-radius if-needed group radius
aaa authorization exec default group radius
aaa authorization network default group radius
```

The lines in this sample RADIUS authentication and authorization configuration are defined as follows:

- The **aaa authentication login use-radius group radius local** command configures the router to use RADIUS for authentication at the login prompt. If RADIUS returns an error, the user is authenticated using the local database. In this example, **use-radius** is the name of the method list, which specifies RADIUS and then local authentication.
- The **aaa authentication ppp user-radius if-needed group radius** command configures the Cisco IOS software to use RADIUS authentication for lines using Point-to-Point Protocol (PPP) with CHAP or PAP if the user has not already been authorized. If the EXEC facility has authenticated the user, RADIUS authentication is not performed. In this example, **user-radius** is the name of the method list defining RADIUS as the if-needed authentication method.
- The **aaa authorization exec default group radius** command sets the RADIUS information that is used for EXEC authorization, autocommands, and access lists.
- The **aaa authorization network default group radius** command sets RADIUS for network authorization, address assignment, and access lists.

RADIUS Authentication, Authorization, and Accounting Example

The following example shows a general configuration using RADIUS with the AAA command set:

```
radius-server host 123.45.1.2
radius-server key myRaDiUSpassWoRd
username root password ALongPassword
aaa authentication ppp dialins group radius local
aaa authorization network default group radius local
aaa accounting network default start-stop group radius
aaa authentication login admins local
aaa authorization exec default local
line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem ri-is-cd
interface group-async 1
 encaps ppp
 ppp authentication pap dialins
```

The lines in this example RADIUS authentication, authorization, and accounting configuration are defined as follows:

- The **radius-server host** command defines the IP address of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list "dialins," which specifies that RADIUS authentication, then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.
- The **ppp authentication pap dialins** command applies the "dialins" method list to the lines specified.
- The **aaa authorization network default group radius local** command is used to assign an address and other network parameters to the RADIUS user.
- The **aaa accounting network default start-stop group radius** command tracks PPP usage.
- The **aaa authentication login admins local** command defines another method list, "admins," for login authentication.
- The **login authentication admins** command applies the "admins" method list for login authentication.

Vendor-Proprietary RADIUS Configuration Example

The following example shows a general configuration using vendor-proprietary RADIUS with the AAA command set:

```
radius-server host alcatraz non-standard
radius-server key myRaDiUSpassWoRd
radius-server configure-nas
username root password ALongPassword
aaa authentication ppp dialins group radius local
aaa authorization network default group radius local
aaa accounting network default start-stop group radius
aaa authentication login admins local
aaa authorization exec default local
line 1 16
autoselect ppp
autoselect during-login
login authentication admins
modem ri-is-cd
interface group-async 1
encaps ppp
ppp authentication pap dialins
```

The lines in this example RADIUS authentication, authorization, and accounting configuration are defined as follows:

- The **radius-server host non-standard** command defines the name of the RADIUS server host and identifies that this RADIUS host uses a vendor-proprietary version of RADIUS.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **radius-server configure-nas** command defines that the Cisco router or access server will query the RADIUS server for static routes and IP pool definitions when the device first starts up.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list "dialins," which specifies that RADIUS authentication, then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.
- The **ppp authentication pap dialins** command applies the "dialins" method list to the lines specified.
- The **aaa authorization network default group radius local** command is used to assign an address and other network parameters to the RADIUS user.
- The **aaa accounting network default start-stop group radius** command tracks PPP usage.
- The **aaa authentication login admins local** command defines another method list, "admins," for login authentication.
- The **login authentication admins** command applies the "admins" method list for login authentication.

RADIUS Server with Server-Specific Values Example

The following example shows how to configure server-specific timeout, retransmit, and key values for the RADIUS server with IP address 172.31.39.46:

```
radius-server host 172.31.39.46 timeout 6 retransmit 5 key rad123
```

Multiple RADIUS Servers with Global and Server-Specific Values Example

The following example shows how to configure two RADIUS servers with specific timeout, retransmit, and key values. In this example, the **aaa new-model** command enables AAA services on the router, while specific AAA commands define the AAA services. The **radius-server retransmit** command changes the global retransmission value to 4 for all RADIUS servers. The **radius-server host** command configures specific timeout, retransmission, and key values for the RADIUS server hosts with IP addresses 172.16.1.1 and 172.29.39.46.

```
! Enable AAA services on the router and define those services.
aaa new-model
aaa authentication login default group radius
aaa authentication login console-login none
aaa authentication ppp default group radius
aaa authorization network default group radius
aaa accounting exec default start-stop group radius
aaa accounting network default start-stop group radius
enable password tryit1
!
! Change the global retransmission value for all RADIUS servers.
radius-server retransmit 4
!
! Configure per-server specific timeout, retransmission, and key values.
! Change the default auth-port and acct-port values.
radius-server host 172.16.1.1 auth-port 1612 acct-port 1616 timeout 3 retransmit 3 key
radkey
!
! Configure per-server specific timeout and key values. This server uses the global
! retransmission value.
radius-server host 172.29.39.46 timeout 6 key rad123
```

Multiple RADIUS Server Entries for the Same Server IP Address Example

The following example shows how to configure the network access server to recognize several RADIUS host entries with the same IP address. Two different host entries on the same RADIUS server are configured for the same services—authentication and accounting. The second host entry configured acts as fail-over backup to the first one. (The RADIUS host entries will be tried in the order they are configured.)

```
! This command enables AAA.
aaa new-model
! The next command configures default RADIUS parameters.
aaa authentication ppp default group radius
! The next set of commands configures multiple host entries for the same IP address.
radius-server host 172.20.0.1 auth-port 1000 acct-port 1001
radius-server host 172.20.0.1 auth-port 2000 acct-port 2000
```

RADIUS Server Group Examples

The following example shows how to create server group *radgroup1* with three different RADIUS server members, each using the default authentication port (1645) and accounting port (1646).

```
aaa group server radius radgroup1
   server 172.16.1.11
   server 172.17.1.21
   server 172.18.1.31
```

The following example shows how to create server group *radgroup2* with three RADIUS server members, each with the same IP address but with unique authentication and accounting ports:

```
aaa group server radius radgroup2
   server 172.16.1.1 auth-port 1000 acct-port 1001
   server 172.16.1.1 auth-port 2000 acct-port 2001
   server 172.16.1.1 auth-port 3000 acct-port 3001
```

Multiple RADIUS Server Entries Using AAA Server Groups Example

The following example shows how to configure the network access server to recognize two different RADIUS server groups. One of these groups, group1, has two different host entries on the same RADIUS server configured for the same services. The second host entry configured acts as fail-over backup to the first one.

```
! This command enables AAA.
aaa new-model
! The next command configures default RADIUS parameters.
aaa authentication ppp default group group1
! The following commands define the group1 RADIUS server group and associate servers
! with it.
aaa group server radius group1
   server 172.20.0.1 auth-port 1000 acct-port 1001
! The following commands define the group2 RADIUS server group and associate servers
! with it.
aaa group server radius group2
   server 172.20.0.1 auth-port 2000 acct-port 2001
! The following set of commands configures the RADIUS attributes for each host entry
! associated with one of the defined server groups.
radius-server host 172.20.0.1 auth-port 1000 acct-port 1001
radius-server host 172.20.0.1 auth-port 2000 acct-port 2001
radius-server host 172.10.0.1 auth-port 1645 acct-port 1646
```

AAA Server Group Selection Based on DNIS Example

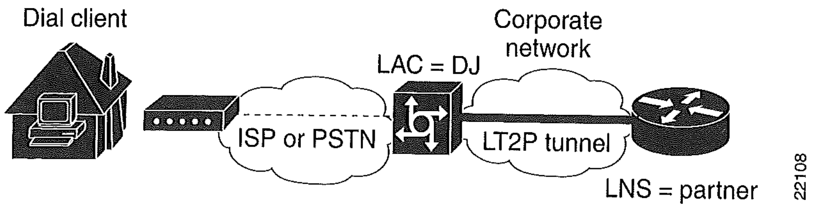
The following example shows how to select RADIUS server groups based on DNIS to provide specific AAA services:

```
! This command enables AAA.
aaa new-model
!
! The following set of commands configures the RADIUS attributes for each server
! that will be associated with one of the defined server groups.
radius-server host 172.16.0.1 auth-port 1645 acct-port 1646 key cisco1
radius-server host 172.17.0.1 auth-port 1645 acct-port 1646 key cisco2
radius-server host 172.18.0.1 auth-port 1645 acct-port 1646 key cisco3
radius-server host 172.19.0.1 auth-port 1645 acct-port 1646 key cisco4
radius-server host 172.20.0.1 auth-port 1645 acct-port 1646 key cisco5

! The following commands define the sg1 RADIUS server group and associate servers
! with it.
aaa group server radius sg1
  server 172.16.0.1
  server 172.17.0.1
! The following commands define the sg2 RADIUS server group and associate a server
! with it.
aaa group server radius sg2
  server 172.18.0.1
! The following commands define the sg3 RADIUS server group and associate a server
! with it.
aaa group server radius sg3
  server 172.19.0.1
! The following commands define the default-group RADIUS server group and associate
! a server with it.
aaa group server radius default-group
  server 172.20.0.1
!
! The next set of commands configures default-group RADIUS server group parameters.
aaa authentication ppp default group default-group
aaa accounting network default start-stop group default-group
!
! The next set of commands enables DNIS mapping and maps DNIS numbers to the defined
! RADIUS server groups. In this configuration, all PPP connection requests using
! DNIS 7777 are sent to the sg1 server group. The accounting records for these
! connections (specifically, start-stop records) are handled by the sg2 server group.
! Calls with a DNIS of 8888 use server group sg3 for authentication and server group
! default-group for accounting. Calls with a DNIS of 9999 use server group
! default-group for authentication and server group sg3 for accounting records
! (stop records only). All other calls with DNIS other than the ones defined use the
! server group default-group for both authentication and stop-start accounting records.
aaa dnis map enable
aaa dnis map 7777 authentication ppp group sg1
aaa dnis map 7777 accounting network start-stop group sg2
aaa dnis map 8888 authentication ppp group sg3
aaa dnis map 9999 accounting network stop-only group sg3
```

L2TP Access Concentrator Examples

The following example shows a basic L2TP configuration for the L2TP access concentrator (LAC) for the topology shown in Figure 8. The local name is not defined, so the host name used is the local name. Because the L2TP tunnel password is not defined, the username password is used. In this example, VPDN is configured locally on the LAC and does not take advantage of the new RADIUS tunnel attributes.

Figure 8 Topology for Configuration Examples



```
! Enable AAA globally.
aaa new-model
! Enable AAA authentication for PPP and list the default method to use for PPP
! authentication.
aaa authentication ppp default local
! Define the username as "DJ."
username DJ password 7 030C5E070A00781B
! Enable VPDN.
vpdn enable
! Define VPDN group number 1.
vpdn-group 1
! Allow the LAC to respond to dialin requests using L2TP from IP address 172.21.9.13
! domain "cisco.com."
request dialin l2tp ip 172.21.9.13 domain cisco.com
```

The following example shows how to configure the LAC if RADIUS tunnel attributes are supported. In this example, there is no local VPDN configuration on the LAC; the LAC, instead, is configured to query the remote RADIUS security server.

```
! Enable global AAA securities services.
aaa new-model
! Enable AAA authentication for PPP and list RADIUS as the default method to use
! for PPP authentication.
aaa authentication ppp default group radius local
! Enable AAA (network) authorization and list RADIUS as the default method to use for
! authorization.
aaa authorization network default group radius
! Define the username as "DJ."
username DJ password 7 030C5E070A00781B
! Enable VPDN.
vpdn enable
! Configure the LAC to interface with the remote RADIUS security server.
radius host 171.69.1.1 auth-port 1645 acct-port 1646
radius-server key cisco
```

L2TP Network Server Example

The following example shows a basic L2TP configuration with corresponding comments on the L2TP network server (LNS) for the topology shown in Figure 8.

```
! Enable AAA globally.
aaa new-model
! Enable AAA authentication for PPP and list the default method to use for PPP
! authentication.
aaa authentication ppp default local
! Define the username as "partner."
username partner password 7 030C5E070A00781B
! Create virtual-template 1 and assign all values for virtual access interfaces.
interface Virtual-Template1
! Borrow the IP address from interface ethernet 1.
 ip unnumbered Ethernet0
! Disable multicast fast switching.
 no ip mroute-cache
! Use CHAP to authenticate PPP.
 ppp authentication chap
! Enable VPDN.
vpdn enable
! Create vpdn-group number 1.
vpdn-group 1
! Accept all dialin l2tp tunnels from virtual-template 1 from remote peer DJ.
accept dialin l2tp virtual-template 1 remote DJ
```

RADIUS User Profile with RADIUS Tunneling Attributes Example

The following example shows a RADIUS user profile (Merit Daemon format) that includes RADIUS tunneling attributes:

```
example.com  Password="cisco" Service-Type=Outbound
Tunnel-Type = :1:L2TP
Tunnel-Medium-Type = :1:IP
Tunnel-Server-Endpoint = :1:10.0.0.1
Tunnel-Password = :1:"welcome"
Tunnel-Assignment-ID = :1:"nas"
```

Configuring TACACS+

This chapter discusses how to enable and configure TACACS+, which provides detailed accounting information and flexible administrative control over authentication and authorization processes. TACACS+ is facilitated through AAA and can be enabled only through AAA commands.

For a complete description of the TACACS+ commands used in this chapter, refer to the "TACACS+ Commands" chapter in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

In This Chapter

This chapter includes the following sections:

- TACACS+ Overview
- TACACS+ Operation
- TACACS+ Configuration Task List
- TACACS+ AV Pairs
- TACACS+ Configuration Examples

TACACS+ Overview

TACACS+ is a security application that provides centralized validation of users attempting to gain access to a router or network access server. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation. You must have access to and must configure a TACACS+ server before the configured TACACS+ features on your network access server are available.

TACACS+ provides for separate and modular authentication, authorization, and accounting facilities. TACACS+ allows for a single access control server (the TACACS+ daemon) to provide each service—authentication, authorization, and accounting—independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

The goal of TACACS+ is to provide a methodology for managing multiple network access points from a single management service. The Cisco family of access servers and routers and the Cisco IOS user interface (for both routers and access servers) can be network access servers.

Network access points enable traditional "dumb" terminals, terminal emulators, workstations, personal computers (PCs), and routers in conjunction with suitable adapters (for example, modems or ISDN adapters) to communicate using protocols such as Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), Compressed SLIP (CSLIP), or AppleTalk Remote Access (ARA) protocol. In other words, a network access server provides connections to a single user, to a network or subnetwork, and to interconnected networks. The entities connected to the network through a network access server are called *network access clients*; for example, a PC running PPP over a voice-grade circuit is a network access client. TACACS+, administered through the AAA security services, can provide the following services:

- Authentication—Provides complete control of authentication through login and password dialog, challenge and response, messaging support.

  The authentication facility provides the ability to conduct an arbitrary dialog with the user (for example, after a login and password are provided, to challenge a user with a number of questions, like home address, mother's maiden name, service type, and social security number). In addition, the TACACS+ authentication service supports sending messages to user screens. For example, a message could notify users that their passwords must be changed because of the company's password aging policy.

- Authorization—Provides fine-grained control over user capabilities for the duration of the user's session, including but not limited to setting autocommands, access control, session duration, or protocol support. You can also enforce restrictions on what commands a user may execute with the TACACS+ authorization feature.

- Accounting—Collects and sends information used for billing, auditing, and reporting to the TACACS+ daemon. Network managers can use the accounting facility to track user activity for a security audit or to provide information for user billing. Accounting records include user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes.

The TACACS+ protocol provides authentication between the network access server and the TACACS+ daemon, and it ensures confidentiality because all protocol exchanges between a network access server and a TACACS+ daemon are encrypted.

You need a system running TACACS+ daemon software to use the TACACS+ functionality on your network access server.

Cisco makes the TACACS+ protocol specification available as a draft RFC for those customers interested in developing their own TACACS+ software.

TACACS+ Operation

When a user attempts a simple ASCII login by authenticating to a network access server using TACACS+, the following process typically occurs:

1 When the connection is established, the network access server will contact the TACACS+ daemon to obtain a username prompt, which is then displayed to the user. The user enters a username and the network access server then contacts the TACACS+ daemon to obtain a password prompt. The network access server displays the password prompt to the user, the user enters a password, and the password is then sent to the TACACS+ daemon.

Note TACACS+ allows an arbitrary conversation to be held between the daemon and the user until the daemon receives enough information to authenticate the user. This is usually done by prompting for a username and password combination, but may include other items, such as mother's maiden name, all under the control of the TACACS+ daemon.

2 The network access server will eventually receive one of the following responses from the TACACS+ daemon:

(a) ACCEPT—The user is authenticated and service may begin. If the network access server is configured to requite authorization, authorization will begin at this time.

(b) REJECT—The user has failed to authenticate. The user may be denied further access, or will be prompted to retry the login sequence depending on the TACACS+ daemon.

(c) ERROR—An error occurred at some time during authentication. This can be either at the daemon or in the network connection between the daemon and the network access server. If an ERROR response is received, the network access server will typically try to use an alternative method for authenticating the user.

(d) CONTINUE—The user is prompted for additional authentication information.

3 A PAP login is similar to an ASCII login, except that the username and password arrive at the network access server in a PAP protocol packet instead of being typed in by the user, so the user is not prompted. PPP CHAP logins are also similar in principle.

Following authentication, the user will also be required to undergo an additional authorization phase, if authorization has been enabled on the network access server. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.

4 If TACACS+ authorization is required, the TACACS+ daemon is again contacted and it returns an ACCEPT or REJECT authorization response. If an ACCEPT response is returned, the response will contain data in the form of attributes that are used to direct the EXEC or NETWORK session for that user, determining services that the user can access.

Services include the following:

(a) Telnet, rlogin, Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services

(b) Connection parameters, including the host or client IP address, access list, and user timeouts

TACACS+ Configuration Task List

To configure your router to support TACACS+, you must perform the following tasks:

- Use the **aaa new-model** global configuration command to enable AAA. AAA must be configured if you plan to use TACACS+. For more information about using the **aaa new-model** command, refer to the "AAA Overview" chapter.
- Use the **tacacs-server host** command to specify the IP address of one or more TACACS+ daemons. Use the **tacacs-server key** command to specify an encryption key that will be used to encrypt all exchanges between the network access server and the TACACS+ daemon. This same key must also be configured on the TACACS+ daemon.
- Use the **aaa authentication** global configuration command to define method lists that use TACACS+ for authentication. For more information about using the **aaa authentication** command, refer to the "Configuring Authentication" chapter.
- Use **line** and **interface** commands to apply the defined method lists to various interfaces. For more information, refer to the "Configuring Authentication" chapter.

- If needed, use the **aaa authorization** global command to configure authorization for the network access server. Unlike authentication, which can be configured per line or per interface, authorization is configured globally for the entire network access server. For more information about using the **aaa authorization** command, refer to the "Configuring Authorization" chapter.
- If needed, use the **aaa accounting** command to enable accounting for TACACS+ connections. For more information about using the **aaa accounting** command, refer to the "Configuring Accounting" chapter.

To configure TACACS+, perform the tasks in the following sections:

- Identifying the TACACS+ Server Host (Required)
- Setting the TACACS+ Authentication Key (Optional)
- Configuring AAA Server Groups (Optional)
- Configuring AAA Server Group Selection Based on DNIS (Optional)
- Specifying TACACS+ Authentication (Required)
- Specifying TACACS+ Authorization (Optional)
- Specifying TACACS+ Accounting (Optional)

For TACACS+ configuration examples using the commands in this chapter, refer to the "TACACS+ Configuration Examples" section at the end of the this chapter.

Identifying the TACACS+ Server Host

The **tacacs-server host** command enables you to specify the names of the IP host or hosts maintaining a TACACS+ server. Because the TACACS+ software searches for the hosts in the order specified, this feature can be useful for setting up a list of preferred daemons.

To specify a TACACS+ host, use the following command in global configuration mode:

Command	Purpose
`tacacs-server host` *`hostname`* `[single-connection] [port` *`integer`*`] [timeout` *`integer`*`] [key` *`string`*`]`	Specifies a TACACS+ host.

Using the **tacacs-server host** command, you can also configure the following options:

- Use the **single-connection** keyword to specify single-connection (only valid with CiscoSecure Release 1.0.1 or later). Rather than have the router open and close a TCP connection to the daemon each time it must communicate, the single-connection option maintains a single open connection between the router and the daemon. This is more efficient because it allows the daemon to handle a higher number of TACACS operations.

Note The daemon must support single-connection mode for this to be effective, otherwise the connection between the network access server and the daemon will lock up or you will receive spurious errors.

- Use the **port** *integer* argument to specify the TCP port number to be used when making connections to the TACACS+ daemon. The default port number is 49.

- Use the **timeout** *integer* argument to specify the period of time (in seconds) the router will wait for a response from the daemon before it times out and declares an error.

Note Specifying the timeout value with the **tacacs-server host** command overrides the default timeout value set with the **tacacs-server timeout** command for this server only.

- Use the **key** *string* argument to specify an encryption key for encrypting and decrypting all traffic between the network access server and the TACACS+ daemon.

Note Specifying the encryption key with the **tacacs-server host** command overrides the default key set by the global configuration **tacacs-server key** command for this server only.

Because some of the parameters of the **tacacs-server host** command override global settings made by the **tacacs-server timeout** and **tacacs-server key** commands, you can use this command to enhance security on your network by uniquely configuring individual TACACS+ connections.

Setting the TACACS+ Authentication Key

To set the global TACACS+ authentication key and encryption key, use the following command in global configuration mode:

Command	Purpose
`tacacs-server key` *key*	Sets the encryption key to match that used on the TACACS+ daemon.

Note You must configure the same key on the TACACS+ daemon for encryption to be successful.

Configuring AAA Server Groups

Configuring the router to use AAA server groups provides a way to group existing server hosts. This allows you to select a subset of the configured server hosts and use them for a particular service. A server group is used in conjunction with a global server-host list. The server group lists the IP addresses of the selected server hosts.

Server groups can include multiple host entries as long as each entry has a unique IP address. If two different host entries in the server group are configured for the same service—for example, accounting—the second host entry configured acts as fail-over backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server will try the second host entry for accounting services. (The TACACS+ host entries will be tried in the order they are configured.)

To define a server host with a server group name, enter the following commands starting in global configuration mode. The listed server must exist in global configuration mode:

Step	Command	Purpose
1	Router(config)# **tacacs-server host** *name* [**single-connection**] [**port** *integer*] [**timeout** *integer*] [**key** *string*]	Specifies and defines the IP address of the server host before configuring the AAA server-group. Refer to the "Identifying the TACACS+ Server Host" section for more information on the **tacacs-server host** command.
2	Router(config-if)# **aaa group server** {**radius** | **tacacs+**} *group-name*	Defines the AAA server-group with a group name. All members of a group must be the same type; that is, RADIUS or TACACS+. This command puts the router in server group subconfiguration mode.
3	Router(config-sg)# **server** *ip-address* [**auth-port** *port number*] [**acct-port** *port-number*]	Associates a particular TACACS+ server with the defined server group. Use the **auth-port** *port-number* option to configure a specific UDP port solely for authentication. Use the **acct-port** *port-number* option to configure a specific UDP port solely for accounting. Repeat this step for each TACACS+ server in the AAA server group. **Note** Each server in the group must be defined previously using the **tacacs-server host** command.

Configuring AAA Server Group Selection Based on DNIS

Cisco IOS allows you to authenticate users to a particular AAA server group based on the session's Dialed Number Identification Service (DNIS) number. Any phone line (a regular home phone or a commercial T1/PRI line) can be associated with several phone numbers. The DNIS number identifies the number that was called to reach you.

For example, suppose you want to share the same phone number with several customers, but you want to know which customer is calling before you pick up the phone. You can customize how you answer the phone because DNIS allows you to know which customer is calling when you answer.

Cisco routers with either ISDN or internal modems can receive the DNIS number. This functionality allows users to assign different TACACS+ server groups for different customers (that is, different TACACS+ servers for different DNIS numbers). Additionally, using server groups you can specify the same server group for AAA services or a separate server group for each AAA service.

Cisco IOS software provides the flexibility to implement authentication and accounting services in several ways:

- Globally—AAA services are defined using global configuration access list commands and applied in general to all interfaces on a specific network access server.
- Per Interface—AAA services are defined using interface configuration commands and applied specifically to the interface being configured on a specific network access server.
- DNIS mapping—You can use DNIS to specify an AAA server to supply AAA services.

Because each of these AAA configuration methods can be configured simultaneously, Cisco has established an order of precedence to determine which server or groups of servers provide AAA services. The order of precedence is as follows:

- Per DNIS—If you configure the network access server to use DNIS to identify/determine which server group provides AAA services, then this method takes precedence over any additional AAA selection method.

- Per interface—If you configure the network access server per interface to use access lists to determine how a server provides AAA services, this method takes precedence over any global configuration AAA access lists.
- Globally—If you configure the network access server by using global AAA access lists to determine how the security server provides AAA services, this method has the least precedence.

Note Prior to configuring AAA Server Group Selection Based on DNIS, you must configure the remote security servers associated with each AAA server group. See the "Identifying the TACACS+ Server Host" and "Configuring AAA Server Groups" sections in this chapter.

To configure the router to select a particular AAA server group based on the DNIS of the server group, configure DNIS mapping. To map a server group with a group name with DNIS number, use the following commands in global configuration mode:

Step	Command	Purpose
1	`Router(config)# aaa dnis map enable`	Enables DNIS mapping.
2	`Router(config)# aaa dnis map dnis-number authentication ppp group server-group-name`	Maps a DNIS number to a defined AAA server group; the servers in this server group are being used for authentication.
3	`Router(config)# aaa dnis map dnis-number accounting network [none | start-stop | stop-only] group server-group-name`	Maps a DNIS number to a defined AAA server group; the servers in this server group are being used for accounting.

Specifying TACACS+ Authentication

After you have identified the TACACS+ daemon and defined an associated TACACS+ encryption key, you need to define method lists for TACACS+ authentication. Because TACACS+ authentication is operated via AAA, you need to issue the **aaa authentication** command, specifying TACACS+ as the authentication method. For more information, refer to the "Configuring Authentication" chapter.

Specifying TACACS+ Authorization

AAA authorization enables you to set parameters that restrict a user's network access. Authorization via TACACS+ may be applied to commands, network connections, and EXEC sessions. Because TACACS+ authorization is facilitated through AAA, you need to issue the **aaa authorization** command, specifying TACACS+ as the authorization method. For more information, refer to the "Configuring Authorization" chapter.

Specifying TACACS+ Accounting

AAA accounting enables you to track the services users are accessing as well as the amount of network resources they are consuming. Because TACACS+ accounting is facilitated through AAA, you need to issue the **aaa accounting** command, specifying TACACS+ as the accounting method. For more information, refer to the "Configuring Accounting" chapter.

TACACS+ AV Pairs

The network access server implements TACACS+ authorization and accounting functions by transmitting and receiving TACACS+ attribute-value (AV) pairs for each user session. For a list of supported TACACS+ AV pairs, refer to the "TACACS+ Attribute-Value Pairs" appendix.

TACACS+ Configuration Examples

The following sections provide TACACS+ configuration examples:

- TACACS+ Authentication Examples
- TACACS+ Authorization Example
- TACACS+ Accounting Example
- TACACS+ Server Group Example
- AAA Server Group Selection Based on DNIS Example
- TACACS+ Daemon Configuration Example

TACACS+ Authentication Examples

The following example shows how to configure TACACS+ as the security protocol for PPP authentication:

```
aaa new-model
aaa authentication ppp test group tacacs+ local
tacacs-server host 10.1.2.3
tacacs-server key goaway
interface serial 0
 ppp authentication chap pap test
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "test," to be used on serial interfaces running PPP. The keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.1.2.3. The **tacacs-server key** command defines the shared encryption key to be "goaway."
- The **interface** command selects the line, and the **ppp authentication** command applies the test method list to this line.

The following example shows how to configure TACACS+ as the security protocol for PPP authentication, but instead of the "test" method list, the "default" method list is used.

```
aaa new-model
aaa authentication ppp default if-needed group tacacs+ local
tacacs-server host 10.1.2.3
tacacs-server key goaway
interface serial 0
 ppp authentication default
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "default," to be used on serial interfaces running PPP. The keyword **default** means that PPP authentication is applied by default to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP authentication is not necessary and can be skipped. If authentication is needed, the keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.1.2.3. The **tacacs-server key** command defines the shared encryption key to be "goaway."
- The **interface** command selects the line, and the **ppp authentication** command applies the default method list to this line.

The following example shows how to create the same authentication algorithm for PAP, but it calls the method list "MIS-access" instead of "default":

```
aaa new-model
aaa authentication pap MIS-access if-needed group tacacs+ local
tacacs-server host 10.1.2.3
tacacs-server key goaway
interface serial 0
 ppp authentication pap MIS-access
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "MIS-access," to be used on serial interfaces running PPP. The method list, "MIS-access," means that PPP authentication is applied to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP authentication is not necessary and can be skipped. If authentication is needed, the keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.1.2.3. The **tacacs-server key** command defines the shared encryption key to be "goaway."
- The **interface** command selects the line, and the **ppp authentication** command applies the default method list to this line.

The following example shows the configuration for a TACACS+ daemon with an IP address of 10.2.3.4 and an encryption key of "apple":

```
aaa new-model
aaa authentication login default group tacacs+ local
tacacs-server host 10.2.3.4
tacacs-server key apple
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines the default method list. Incoming ASCII logins on all interfaces (by default) will use TACACS+ for authentication. If no TACACS+ server responds, then the network access server will use the information contained in the local username database for authentication.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.2.3.4. The **tacacs-server key** command defines the shared encryption key to be "apple."

TACACS+ Authorization Example

The following example shows how to configure TACACS+ as the security protocol for PPP authentication using the default method list; it also shows how to configure network authorization via TACACS+:

```
aaa new-model
aaa authentication ppp default if-needed group tacacs+ local
aaa authorization network default group tacacs+
tacacs-server host 10.1.2.3
tacacs-server key goaway
interface serial 0
 ppp authentication default
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "default," to be used on serial interfaces running PPP. The keyword **default** means that PPP authentication is applied by default to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP authentication is not necessary and can be skipped. If authentication is needed, the keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **aaa authorization** command configures network authorization via TACACS+. Unlike authentication lists, this authorization list always applies to all incoming network connections made to the network access server.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.1.2.3. The **tacacs-server key** command defines the shared encryption key to be "goaway."
- The **interface** command selects the line, and the **ppp authentication** command applies the default method list to this line.

TACACS+ Accounting Example

The following example shows how to configure TACACS+ as the security protocol for PPP authentication using the default method list; it also shows how to configure accounting via TACACS+:

```
aaa new-model
aaa authentication ppp default if-needed group tacacs+ local
aaa accounting network default stop-only group tacacs+
tacacs-server host 10.1.2.3
tacacs-server key goaway
interface serial 0
 ppp authentication default
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "default," to be used on serial interfaces running PPP. The keyword **default** means that PPP authentication is applied by default to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP authentication is not necessary and can be skipped. If authentication is needed, the keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **aaa accounting** command configures network accounting via TACACS+. In this example, accounting records describing the session that just terminated will be sent to the TACACS+ daemon whenever a network connection terminates.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.1.2.3. The **tacacs-server key** command defines the shared encryption key to be "goaway."
- The **interface** command selects the line, and the **ppp authentication** command applies the default method list to this line.

TACACS+ Server Group Example

The following example shows how to create a server group with three different TACACS+ servers members:

```
aaa group server tacacs tacgroup1
   server 172.16.1.1
   server 172.16.1.21
   server 172.16.1.31
```

AAA Server Group Selection Based on DNIS Example

The following example shows how to select TACAC+ server groups based on DNIS to provide specific AAA services:

```
! This command enables AAA.
aaa new-model
!
! The following set of commands configures the TACACS+ servers that will be associated
! with one of the defined server groups.
tacacs-server host 172.16.0.1
tacacs-server host 172.17.0.1
tacacs-server host 172.18.0.1
tacacs-server host 172.19.0.1
tacacs-server host 172.20.0.1
tacacs-server key abcdefg

! The following commands define the sg1 TACACS+ server group and associate servers
! with it.
aaa group server tacacs sg1
  server 172.16.0.1
  server 172.17.0.1
! The following commands define the sg2 TACACS+ server group and associate a server
! with it.
aaa group server tacacs sg2
  server 172.18.0.1
! The following commands define the sg3 TACACS+ server group and associate a server
! with it.
aaa group server tacacs sg3
  server 172.19.0.1
! The following commands define the default-group TACACS+ server group and associate
! a server with it.
aaa group server tacacs default-group
  server 172.20.0.1
!
! The next set of commands configures default-group tacacs server group parameters.
aaa authentication ppp default group default-group
aaa accounting network default start-stop group default-group
!
! The next set of commands enables DNIS mapping and maps DNIS numbers to the defined
! RADIUS server groups. In this configuration, all PPP connection requests using DNIS
! 7777 are sent to the sg1 server group. The accounting records for these connections
! (specifically, start-stop records) are handled by the sg2 server group. Calls with a
! DNIS of 8888 use server group sg3 for authentication and server group default-group
! for accounting. Calls with a DNIS of 9999 use server group default-group for
! authentication and server group sg3 for accounting records (stop records only). All
! other calls with DNIS other than the ones defined use the server group default-group
! for both authentication and stop-start accounting records.
aaa dnis map enable
aaa dnis map 7777 authentication ppp group sg1
aaa dnis map 7777 accounting network start-stop group sg2
aaa dnis map 8888 authentication ppp group sg3
aaa dnis map 9999 accounting network stop-only group sg3
```

TACACS+ Daemon Configuration Example

The following example shows a sample configuration of the TACACS+ daemon. The precise syntax used by your TACACS+ daemon may be different than that included in this example.

```
user = mci_customer1 {
 chap = cleartext "some chap password"
 service = ppp protocol = ip {
inacl#1="permit ip any any precedence immediate"
inacl#2="deny igrp 0.0.1.2 255.255.0.0 any"
 }
}
```

Configuring Kerberos

This chapter describes the Kerberos security system. For a complete description of the Kerberos commands used in this chapter, refer to the "Kerberos Commands" chapter in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

In This Chapter

This chapter includes the following topics and tasks:

- Kerberos Overview
- Kerberos Client Support Operation
- Kerberos Configuration Task List
- Kerberos Configuration Examples

Kerberos Overview

Kerberos is a secret-key network authentication protocol, developed at Massachusetts Institute of Technology (MIT), that uses the Data Encryption Standard (DES) cryptographic algorithm for encryption and authentication. Kerberos was designed to authenticate requests for network resources. Kerberos, like other secret-key systems, is based on the concept of a trusted third party that performs secure verification of users and services. In the Kerberos protocol, this trusted third party is called the key distribution center (KDC).

The primary use of Kerberos is to verify that users and the network services they use are really who and what they claim to be. To accomplish this, a trusted Kerberos server issues tickets to users. These tickets, which have a limited lifespan, are stored in a user's credential cache and can be used in place of the standard username-and-password authentication mechanism.

The Kerberos credential scheme embodies a concept called "single logon." This process requires authenticating a user once, and then allows secure authentication (without encrypting another password) wherever that user's credential is accepted.

Starting with Cisco IOS Release 11.2, Cisco IOS software includes Kerberos 5 support, which allows organizations already deploying Kerberos 5 to use the same Kerberos authentication database on their routers that they are already using on their other network hosts (such as UNIX servers and PCs).

The following network services are supported by the Kerberos authentication capabilities in Cisco IOS software:

- Telnet
- rlogin
- rsh
- rcp

Note Cisco's implementation of Kerberos client support is based on code developed by CyberSafe, which was derived from the MIT code. As a result, the Cisco Kerberos implementation has successfully undergone full compatibility testing with the CyberSafe Challenger commercial Kerberos server and MIT's server code, which is freely distributed.

Table 12 lists common Kerberos-related terms and their definitions.

Table 12 Kerberos Terminology

Term	Definition
Authentication	A process by which a user or service identifies itself to another service. For example, a client can authenticate to a router or a router can authenticate to another router.
Authorization	A means by which the router determines what privileges you have in a network or on the router and what actions you can perform.
Credential	A general term that refers to authentication tickets, such as ticket granting tickets (TGTs) and service credentials. Kerberos credentials verify the identity of a user or service. If a network service decides to trust the Kerberos server that issued a ticket, it can be used in place of retyping in a username and password. Credentials have a default lifespan of eight hours.
Instance	An authorization level label for Kerberos principals. Most Kerberos principals are of the form user@REALM (for example, smith@EXAMPLE.COM). A Kerberos principal with a Kerberos instance has the form user/instance@REALM (for example, smith/admin@EXAMPLE.COM). The Kerberos instance can be used to specify the authorization level for the user if authentication is successful. It is up to the server of each network service to implement and enforce the authorization mappings of Kerberos instances. Note that the Kerberos realm name must be in uppercase characters.
Kerberized	Applications and services that have been modified to support the Kerberos credential infrastructure.
Kerberos realm	A domain consisting of users, hosts, and network services that are registered to a Kerberos server. The Kerberos server is trusted to verify the identity of a user or network service to another user or network service. Kerberos realms must always be in uppercase characters.
Kerberos server	A daemon running on a network host. Users and network services register their identity with the Kerberos server. Network services query the Kerberos server to authenticate to other network services.
Key distribution center (KDC)	A Kerberos server and database program running on a network host.
Principal	Also known as a Kerberos identity, this is who you are or what a service is according to the Kerberos server.
Service credential	A credential for a network service. When issued from the KDC, this credential is encrypted with the password shared by the network service and the KDC, and with the user's TGT.

Table 12 Kerberos Terminology (continued)

Term	Definition
SRVTAB	A password that a network service shares with the KDC. The network service authenticates an encrypted service credential by using the SRVTAB (also known as a KEYTAB) to decrypt it.
Ticket granting ticket (TGT)	A credential that the key distribution center (KDC) issues to authenticated users. When users receive a TGT, they can authenticate to network services within the Kerberos realm represented by the KDC.

Kerberos Client Support Operation

This section describes how the Kerberos security system works with a Cisco router functioning as the security server. Although (for convenience or technical reasons) you can customize Kerberos in a number of ways, remote users attempting to access network services must pass through three layers of security before they can access network services.

This section includes the following sections:

- Authenticating to the Boundary Router
- Obtaining a TGT from a KDC
- Authenticating to Network Services

Authenticating to the Boundary Router

This section describes the first layer of security that remote users must pass through when they attempt to access a network. The first step in the Kerberos authentication process is for users to authenticate themselves to the boundary router. The following process describes how users authenticate to a boundary router:

1. The remote user opens a PPP connection to the corporate site router.
2. The router prompts the user for a username and password.
3. The router requests a TGT from the KDC for this particular user.
4. The KDC sends an encrypted TGT to the router that includes (among other things) the user's identity.
5. The router attempts to decrypt the TGT using the password the user entered. If the decryption is successful, the remote user is authenticated to the router.

A remote user who successfully initiates a PPP session and authenticates to the boundary router is inside the firewall but still must authenticate to the KDC directly before being allowed to access network services. This is because the TGT issued by the KDC is stored on the router and is not useful for additional authentication unless the user physically logs on to the router.

Obtaining a TGT from a KDC

This section describes how remote users who are authenticated to the boundary router authenticate themselves to a KDC.

When a remote user authenticates to a boundary router, that user technically becomes part of the network; that is, the network is extended to include the remote user and the user's machine or network. To gain access to network services, however, the remote user must obtain a TGT from the KDC. The following process describes how remote users authenticate to the KDC:

1. The remote user, at a workstation on a remote site, launches the KINIT program (part of the client software provided with the Kerberos protocol).
2. The KINIT program finds the user's identity and requests a TGT from the KDC.
3. The KDC creates a TGT, which contains the identity of the user, the identity of the KDC, and the TGT's expiration time.
4. Using the user's password as a key, the KDC encrypts the TGT and sends the TGT to the workstation.
5. When the KINIT program receives the encrypted TGT, it prompts the user for a password (this is the password that is defined for the user in the KDC).
6. If the KINIT program can decrypt the TGT with the password the user enters, the user is authenticated to the KDC, and the KINIT program stores the TGT in the user's credential cache.

At this point, the user has a TGT and can communicate securely with the KDC. In turn, the TGT allows the user to authenticate to other network services.

Authenticating to Network Services

The following process describes how a remote user with a TGT authenticates to network services within a given Kerberos realm. Assume the user is on a remote workstation (Host A) and wants to log in to Host B.

1. The user on Host A initiates a Kerberized application (such as Telnet) to Host B.
2. The Kerberized application builds a service credential request and sends it to the KDC. The service credential request includes (among other things) the user's identity and the identity of the desired network service. The TGT is used to encrypt the service credential request.
3. The KDC tries to decrypt the service credential request with the TGT it issued to the user on Host A. If the KDC can decrypt the packet, it is assured that the authenticated user on Host A sent the request.
4. The KDC notes the network service identity in the service credential request.
5. The KDC builds a service credential for the appropriate network service on Host B on behalf of the user on Host A. The service credential contains the client's identity and the desired network service's identity.
6. The KDC then encrypts the service credential twice. It first encrypts the credential with the SRVTAB that it shares with the network service identified in the credential. It then encrypts the resulting packet with the TGT of the user (who, in this case, is on Host A).
7. The KDC sends the twice-encrypted credential to Host A.
8. Host A attempts to decrypt the service credential with the user's TGT. If Host A can decrypt the service credential, it is assured the credential came from the real KDC.

9 Host A sends the service credential to the desired network service. Note that the credential is still encrypted with the SRVTAB shared by the KDC and the network service.

10 The network service attempts to decrypt the service credential using its SRVTAB.

11 If the network service can decrypt the credential, it is assured the credential was in fact issued from the KDC. Note that the network service trusts anything it can decrypt from the KDC, even if it receives it indirectly from a user. This is because the user first authenticated with the KDC.

At this point, the user is authenticated to the network service on Host B. This process is repeated each time a user wants to access a network service in the Kerberos realm.

Kerberos Configuration Task List

For hosts and the KDC in your Kerberos realm to communicate and mutually authenticate, you must identify them to each other. To do this, you add entries for the hosts to the Kerberos database on the KDC and add SRVTAB files generated by the KDC to all hosts in the Kerberos realm. You also make entries for users in the KDC database.

This section describes how to set up a Kerberos-authenticated server-client system and contains the following topics:

- Configuring the KDC Using Kerberos Commands
- Configuring the Router to Use the Kerberos Protocol

This section assumes that you have installed the Kerberos administrative programs on a UNIX host, known as the KDC, initialized the database, and selected a Kerberos realm name and password. For instructions about completing these tasks, refer to documentation that came with your Kerberos software.

Note Write down the host name or IP address of the KDC, the port number you want the KDC to monitor for queries, and the name of the Kerberos realm it will serve. You need this information to configure the router.

Configuring the KDC Using Kerberos Commands

After you set up a host to function as the KDC in your Kerberos realm, you must make entries to the KDC database for all principals in the realm. Principals can be network services on Cisco routers and hosts or they can be users.

To use Kerberos commands to add services to the KDC database (and to modify existing database information), complete the tasks in the following sections:

- Adding Users to the KDC Database
- Creating SRVTABs on the KDC
- Extracting SRVTABs

Note All Kerberos command examples are based on Kerberos 5 Beta 5 of the original MIT implementation. Later versions use a slightly different interface.

Adding Users to the KDC Database

To add users to the KDC and create privileged instances of those users, use the **su** command to become root on the host running the KDC and use the kdb5_edit program to use the following commands:

Step	Command	Purpose
1	**ank** *username@REALM*	Use the **ank** (add new key) command to add a user to the KDC. This command prompts for a password, which the user must enter to authenticate to the router.
2	**ank** *username/instance@REALM*	Use the **ank** command to add a privileged instance of a user.

For example, to add user *loki* of Kerberos realm CISCO.COM, enter the following Kerberos command:

```
ank loki@CISCO.COM
```

Note The Kerberos realm name must be in uppercase characters.

You might want to create privileged instances to allow network administrators to connect to the router at the enable level, for example, so that they need not enter a clear text password (and compromise security) to enter enable mode.

To add an instance of *loki* with additional privileges (in this case, *enable*, although it could be anything) enter the following Kerberos command:

```
ank loki/enable@CISCO.COM
```

In each of these examples, you are prompted to enter a password, which you must give to user *loki* to use at login.

The "Enabling Kerberos Instance Mapping" section describes how to map Kerberos instances to various Cisco IOS privilege levels.

Creating SRVTABs on the KDC

All routers that you want to authenticate to use the Kerberos protocol must have an SRVTAB. This section and the "Extracting SRVTABs" section describe how to create and extract SRVTABs for a router called *router1*. The section "Copying SRVTAB Files" describes how to copy SRVTAB files to the router.

To make SRVTAB entries on the KDC, use the following command:

Command	Purpose
ark *SERVICE/HOSTNAME@REALM*	Use the **ark** (add random key) command to add a network service supported by a host or router to the KDC.

For example, to add a Kerberized authentication service for a Cisco router called *router1* to the Kerberos realm CISCO.COM, enter the following Kerberos command:

```
ark host/router1.cisco.com@CISCO.COM
```

Make entries for all network services on all Kerberized hosts that use this KDC for authentication.

Extracting SRVTABs

SRVTABs contain (among other things) the passwords or randomly generated keys for the service principals you entered into the KDC database. Service principal keys must be shared with the host running that service. To do this, you must save the SRVTAB entries to a file, then copy the file to the router and all hosts in the Kerberos realm. Saving SRVTAB entries to a file is called *extracting* SRVTABs. To extract SRVTABs, use the following command:

Command	Purpose
xst *router-name host*	Use the kdb5_edit command **xst** to write an SRVTAB entry to a file.

For example, to write the host/router1.cisco.com@CISCO.COM SRVTAB to a file, enter the following Kerberos command:

```
xst router1.cisco.com@CISCO.COM host
```

Use the **quit** command to exit the kdb5_edit program.

Configuring the Router to Use the Kerberos Protocol

To configure a Cisco router to function as a network security server and authenticate users using the Kerberos protocol, complete the tasks in the following sections:

- Defining a Kerberos Realm
- Copying SRVTAB Files
- Specifying Kerberos Authentication
- Enabling Credentials Forwarding
- Telneting to the Router
- Establishing an Encrypted Kerberized Telnet Session
- Enabling Mandatory Kerberos Authentication
- Enabling Kerberos Instance Mapping
- Monitoring and Maintaining Kerberos

Defining a Kerberos Realm

For a router to authenticate a user defined in the Kerberos database, it must know the host name or IP address of the host running the KDC, the name of the Kerberos realm and, optionally, be able to map the host name or Domain Name System (DNS) domain to the Kerberos realm.

To configure the router to authenticate to a specified KDC in a specified Kerberos realm, use the following commands in global configuration mode. Note that DNS domain names must begin with a leading dot (.):

Step	Command	Purpose
1	**kerberos local-realm** *kerberos-realm*	Defines the default realm for the router.
2	**kerberos server** *kerberos-realm* {*hostname* | *ip-address*} [*port-number*]	Specifies to the router which KDC to use in a given Kerberos realm and, optionally, the port number the KDC is monitoring. (The default is 88.)
3	**kerberos realm** {*dns-domain* | *host*} *kerberos-realm*	(Optional) Maps a host name or DNS domain to a Kerberos realm.

Note Because the machine running the KDC and all Kerberized hosts must interact within a 5-minute window or authentication fails, all Kerberized machines, and especially the KDC, should be running the Network Time Protocol (NTP).

The **kerberos local-realm**, **kerberos realm**, and **kerberos server** commands are equivalent to the UNIX *krb.conf* file. Table 13 identifies mappings from the Cisco IOS configuration commands to a Kerberos 5 configuration file (krb5.conf).

Table 13 Kerberos 5 Configuration File and Commands

krb5.conf File	Cisco IOS Configuration Command
[libdefaults] default_realm = *DOMAIN.COM*	(in config mode) **kerberos local-realm** *DOMAIN.COM*
[domain_realm] .domain.com = *DOMAIN.COM* domain.com = *DOMAIN.COM*	(in config mode) **kerberos realm** *.domain.com DOMAIN.COM* **kerberos realm** *domain.com DOMAIN.COM*

[realms]
kdc = *DOMAIN.PIL.COM:750*
admin_server = *DOMAIN.PIL.COM*
default_domain = *DOMAIN.COM*
	(in config mode)
	kerberos server *DOMAIN.COM 172.65.44.2*
	(*172.65.44.2* is the example IP address for *DOMAIN.PIL.COM*)

For an example of defining a Kerberos realm, see the "Defining a Kerberos Realm" section at the end of this chapter.

Copying SRVTAB Files

To make it possible for remote users to authenticate to the router using Kerberos credentials, the router must share a secret key with the KDC. To do this, you must give the router a copy of the SRVTAB you extracted on the KDC.

The most secure method to copy SRVTAB files to the hosts in your Kerberos realm is to copy them onto physical media and go to each host in turn and manually copy the files onto the system. To copy SRVTAB files to the router, which does not have a physical media drive, you must transfer them via the network using the Trivial File Transfer Protocol (TFTP).

To remotely copy SRVTAB files to the router from the KDC, use the following command in global configuration mode:

Command	Purpose
kerberos srvtab remote {*hostname* | *ip-address*} {*filename*}	Retrieves an SRVTAB file from the KDC.

When you copy the SRVTAB file from the router to the KDC, the **kerberos srvtab remote** command parses the information in this file and stores it in the router's running configuration in the **kerberos srvtab entry** format. To ensure that the SRVTAB is available (does not need to be acquired from the KDC) when you reboot the router, use the **write memory** configuration command to write your running configuration (which contains the parsed SRVTAB file) to NVRAM.

For an example of copying SRVTAB files, see the "SRVTAB File Copying Example" section at the end of this chapter.

Specifying Kerberos Authentication

You have now configured Kerberos on your router. This makes it possible for the router to authenticate using Kerberos. The next step is to tell it to do so. Because Kerberos authentication is facilitated through AAA, you need to enter the **aaa authentication** command, specifying Kerberos as the authentication method. For more information, refer to the "Configuring Authentication" chapter.

Enabling Credentials Forwarding

With Kerberos configured thus far, a user authenticated to a Kerberized router has a TGT and can use it to authenticate to a host on the network. However, if the user tries to list credentials after authenticating to a host, the output will show no Kerberos credentials present.

You can optionally configure the router to forward users' TGTs with them as they authenticate from the router to Kerberized remote hosts on the network when using Kerberized Telnet, rcp, rsh, and rlogin (with the appropriate flags).

To force all clients to forward users' credentials as they connect to other hosts in the Kerberos realm, use the following command in global configuration mode:

Command	Purpose
kerberos credential forward	Forces all clients to forward user credentials upon successful Kerberos authentication.

With credentials forwarding enabled, users' TGTs are automatically forwarded to the next host they authenticate to. In this way, users can connect to multiple hosts in the Kerberos realm without running the KINIT program each time to get a new TGT.

Telneting to the Router

To use Kerberos to authenticate users opening a Telnet session to the router from within the network, use the following command in global configuration mode:

Command	Purpose
aaa authentication login {**default** | *list-name*} **krb5_telnet**	Sets login authentication to use the Kerberos 5 Telnet authentication protocol when using Telnet to connect to the router.

Although Telnet sessions to the router are authenticated, users must still enter a clear text password if they want to enter enable mode. The **kerberos instance map** command, discussed in a later section, allows them to authenticate to the router at a predefined privilege level.

Establishing an Encrypted Kerberized Telnet Session

Another way for users to open a secure Telnet session is to use Encrypted Kerberized Telnet. With Encrypted Kerberized Telnet, users are authenticated by their Kerberos credentials before a Telnet session is established. The Telnet session is encrypted using 56-bit Data Encryption Standard (DES) encryption with 64-bit Cipher Feedback (CFB). Because data sent or received is encrypted, not clear text, the integrity of the dialed router or access server can be more easily controlled.

Note This feature is available only if you have the 56-bit encryption image. 56-bit DES encryption is subject to U.S. Government export control regulations.

To establish an encrypted Kerberized Telnet session from a router to a remote host, use either of the following commands in EXEC command mode:

Command	Purpose
connect *host* [*port*] **/encrypt kerberos** or **telnet** *host* [*port*] **/encrypt kerberos**	Establishes an encrypted Telnet session.

When a user opens a Telnet session from a Cisco router to a remote host, the router and remote host negotiate to authenticate the user using Kerberos credentials. If this authentication is successful, the router and remote host then negotiate whether or not to use encryption. If this negotiation is successful, both inbound and outbound traffic is encrypted using 56-bit DES encryption with 64-bit CFB.

When a user dials in from a remote host to a Cisco router configured for Kerberos authentication, the host and router will attempt to negotiate whether or not to use encryption for the Telnet session. If this negotiation is successful, the router will encrypt all outbound data during the Telnet session.

If encryption is not successfully negotiated, the session will be terminated and the user will receive a message stating that the encrypted Telnet session was not successfully established.

For information about enabling bidirectional encryption from a remote host, refer to the documentation specific to the remote host device.

For an example of using encrypted Kerberized Telnet to open a secure Telnet session, see the "Encrypted Telnet Session Example" section at the end of this chapter.

Enabling Mandatory Kerberos Authentication

As an added layer of security, you can optionally configure the router so that, after remote users authenticate to it, these users can authenticate to other services on the network only with Kerberized Telnet, rlogin, rsh, and rcp. If you do not make Kerberos authentication mandatory and Kerberos authentication fails, the application attempts to authenticate users using the default method of authentication for that network service; for example, Telnet and rlogin prompt for a password, and rsh attempts to authenticate using the local rhost file.

To make Kerberos authentication mandatory, use the following command in global configuration mode:

Command	Purpose
kerberos clients mandatory	Sets Telnet, rlogin, rsh, and rcp to fail if they cannot negotiate the Kerberos protocol with the remote server.

Enabling Kerberos Instance Mapping

As mentioned in the section "Creating SRVTABs on the KDC," you can create administrative instances of users in the KDC database. The **kerberos instance map** command allows you to map those instances to Cisco IOS privilege levels so that users can open secure Telnet sessions to the router at a predefined privilege level, obviating the need to enter a clear text password to enter enable mode.

To map a Kerberos instance to a Cisco IOS privilege level, use the following command in global configuration mode:

Command	Purpose
kerberos instance map *instance privilege-level*	Maps a Kerberos instance to a Cisco IOS privilege level.

If there is a Kerberos instance for user *loki* in the KDC database (for example, *loki/admin*), user *loki* can now open a Telnet session to the router as loki/admin and authenticate automatically at privilege level 15, assuming instance "admin" is mapped to privilege level 15. (See the section "Adding Users to the KDC Database" earlier in this chapter.)

Cisco IOS commands can be set to various privilege levels using the **privilege level** command.

After you map a Kerberos instance to a Cisco IOS privilege level, you must configure the router to check for Kerberos instances each time a user logs in. To run authorization to determine if a user is allowed to run an EXEC shell based on a mapped Kerberos instance, use the **aaa authorization** command with the **krb5-instance** keyword. For more information, refer to the "Configuring Authorization" chapter.

Monitoring and Maintaining Kerberos

To display or remove a current user's credentials, use the following commands in EXEC mode:

Step	Command	Purpose
1	**show kerberos creds**	Lists the credentials in a current user's credentials cache.
2	**clear kerberos creds**	Destroys all credentials in a current user's credentials cache.

For an example of Kerberos configuration, see the "Kerberos Configuration Examples" section.

Kerberos Configuration Examples

The following sections provide Kerberos configuration examples:

- Kerberos Realm Definition Examples
- SRVTAB File Copying Example
- Kerberos Configuration Examples
- Encrypted Telnet Session Example

Kerberos Realm Definition Examples

To define CISCO.COM as the default Kerberos realm, use the following command:

```
kerberos local-realm CISCO.COM
```

To tell the router that the CISCO.COM KDC is running on host 10.2.3.4 at port number 170, use the following Kerberos command:

```
kerberos server CISCO.COM 10.2.3.4 170
```

To map the DNS domain cisco.com to the Kerberos realm CISCO.COM, use the following command:

```
kerberos realm .cisco.com CISCO.COM
```

SRVTAB File Copying Example

To copy over the SRVTAB file on a host named host123.cisco.com for a router named router1.cisco.com, the command would look like this:

```
kerberos srvtab remote host123.cisco.com router1.cisco.com-new-srvtab
```

Kerberos Configuration Examples

This section provides a typical non-Kerberos router configuration and shows output for this configuration from the **write term** command, then builds on this configuration by adding optional Kerberos functionality. Output for each configuration is presented for comparison against the previous configuration.

This example shows how to use the kdb5_edit program to perform the following configuration tasks:

- Adding user chet to the Kerberos database
- Adding a privileged Kerberos instance of user chet (chet/admin) to the Kerberos database
- Adding a restricted instance of chet (chet/restricted) to the Kerberos database
- Adding workstation chet-ss20.cisco.com
- Adding router chet-2500.cisco.com to the Kerberos database
- Adding workstation chet-ss20.cisco.com to the Kerberos database
- Extracting SRVTABs for the router and workstations
- Listing the contents of the KDC database (with the **ldb** command)

Note that, in this sample configuration, host chet-ss20 is also the KDC:

```
chet-ss20# sbin/kdb5_edit
kdb5_edit:  ank chet
Enter password:
Re-enter password for verification:
kdb5_edit:  ank chet/admin
Enter password:
Re-enter password for verification:
kdb5_edit:  ank chet/restricted
Enter password:
Re-enter password for verification:
kdb5_edit:  ark host/chet-ss20.cisco.com
kdb5_edit:  ark host/chet-2500.cisco.com
kdb5_edit:  xst chet-ss20.cisco.com host
'host/chet-ss20.cisco.com@CISCO.COM' added to keytab
'WRFILE:chet-ss20.cisco.com-new-srvtab'
kdb5_edit:  xst chet-2500.cisco.com host
'host/chet-2500.cisco.com@CISCO.COM' added to keytab
'WRFILE:chet-2500.cisco.com-new-srvtab'
kdb5_edit:  ldb
entry: host/chet-2500.cisco.com@CISCO.COM
entry: chet/restricted@CISCO.COM
entry: chet@CISCO.COM
entry: K/M@CISCO.COM
entry: host/chet-ss20.cisco.com@CISCO.COM
entry: krbtgt/CISCO.COM@CISCO.COM
entry: chet/admin@CISCO.COM
kdb5_edit:  q
chet-ss20#
```

The following example shows output from a **write term** command, which displays the configuration of router chet-2500. This is a typical configuration with no Kerberos authentication.

```
chet-2500# write term
Building configuration...

Current configuration:
!
! Last configuration
change at 14:03:55 PDT Mon May 13 1996
!
version 11.2
service udp-small-servers
service tcp-small-servers
!
hostname chet-2500
!
clock timezone PST -8
clock summer-time PDT recurring
aaa new-model
aaa authentication login console none
aaa authentication ppp local local
enable password sMudgKin
!
username chet-2500 password 7 sMudgkin
username chet-3000 password 7 sMudgkin
username chetin password 7 sMudgkin
!
interface Ethernet0
 ip address 172.16.0.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
 no fair-queue
!
interface Serial1
 no ip address
 shutdown
 no fair-queue
!
interface Async2
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
interface Async3
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic address
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
router eigrp 109
 network 172.17.0.0
 no auto-summary
!
ip default-gateway 172.30.55.64
ip domain-name cisco.com
ip name-server 192.168.0.0
ip classless
!
!

line con 0
 exec-timeout 0 0
 login authentication console
line 1 16
 transport input all
line aux 0
 transport input all
line vty 0 4
 password sMudgKin
!
ntp clock-period 17179703
ntp peer 172.19.10.0
ntp peer 172.19.0.0
end
```

The following example shows how to enable user authentication on the router via the Kerberos database. To enable user authentication via the Kerberos database, you would perform the following tasks:

- Entering configuration mode
- Defining the Kerberos local realm
- Identifying the machine hosting the KDC
- Enabling credentials forwarding

- Specifying Kerberos as the method of authentication for login
- Exiting configuration mode (CTL-Z)
- Writing the new configuration to the terminal

```
chet-2500# configure term
Enter configuration commands, one per line.  End with CNTL/Z.
chet-2500(config)# kerberos local-realm CISCO.COM
chet-2500(config)# kerberos server CISCO.COM chet-ss20
Translating "chet-ss20"...domain server (192.168.0.0) [OK]

chet-2500(config)# kerberos credentials forward
chet-2500(config)# aaa authentication login default krb5
chet-2500(config)#
chet-2500#
%SYS-5-CONFIG_I: Configured from console by console
chet-2500# write term
```

Compare the following configuration with the previous one. In particular, look at the lines beginning with the words "aaa," "username," and "kerberos" (lines 10 through 20) in this new configuration.

```
Building configuration...

Current configuration:
!
! Last configuration change at 14:05:54 PDT Mon May 13 1996
!
version 11.2
service udp-small-servers
service tcp-small-servers
!
hostname chet-2500
!
clock timezone PST -8
clock summer-time PDT recurring
aaa new-model
aaa authentication login default krb5
aaa authentication login console none
aaa authentication ppp local local
enable password sMudgKin
!
username chet-2500 password 7 sMudgkin
username chet-3000 password 7 sMudgkin
username chetin password 7 sMudgkin
kerberos local-realm CISCO.COM
kerberos server CISCO.COM 172.71.54.14
kerberos credentials forward
!
interface Ethernet0
 ip address 172.16.0.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
 no fair-queue
!
interface Serial1
 no ip address
 shutdown
 no fair-queue
!
interface Async2
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
interface Async3
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic address
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
router eigrp 109
 network 172.17.0.0
 no auto-summary
!
ip default-gateway 172.30.55.64
ip domain-name cisco.com
ip name-server 192.168.0.0
ip classless
!
!
line con 0
 exec-timeout 0 0
 login authentication console
line 1 16
 transport input all
line aux 0
 transport input all
line vty 0 4
 password sMudgKin
!
ntp clock-period 17179703
ntp peer 172.19.10.0
ntp peer 172.19.0.0
end
```

With the router configured thus far, user chet can log in to the router with a username and password and automatically obtain a TGT, as illustrated in the next example. With possession of a credential, user chet successfully authenticates to host chet-ss20 without entering a username/password.

```
chet-ss20% telnet chet-2500
Trying 172.16.0.0 ...
Connected to chet-2500.cisco.com.
Escape character is '^]'.


User Access Verification

Username: chet
Password:

chet-2500> show kerberos creds
Default Principal:  chet@CISCO.COM
Valid Starting          Expires                 Service Principal
13-May-1996 14:05:39    13-May-1996 22:06:40    krbtgt/CISCO.COM@CISCO.COM

chet-2500> telnet chet-ss20
```

```
Trying chet-ss20.cisco.com (172.71.54.14)... Open
Kerberos:       Successfully forwarded credentials


SunOS UNIX (chet-ss20) (pts/7)

Last login: Mon May 13 13:47:35 from chet-ss20.cisco.c
Sun Microsystems Inc.   SunOS 5.4       Generic July 1994
unknown mode: new
chet-ss20%
```

The following example shows how to authenticate to the router using Kerberos credentials. To authenticate using Kerberos credentials, you would perform the following tasks:

- Entering configuration mode
- Remotely copying over the SRVTAB file from the KDC
- Setting authentication at login to use the Kerberos 5 Telnet authentication protocol when using Telnet to connect to the router
- Writing the configuration to the terminal

Note that the new configuration contains a **kerberos srvtab entry** line. This line is created by the **kerberos srvtab remote** command.

```
chet-2500# configure term
Enter configuration commands, one per line.  End with CNTL/Z.
chet-2500(config)#kerberos srvtab remote earth chet/chet-2500.cisco.com-new-srvtab
Translating "earth"...domain server (192.168.0.0) [OK]

Loading chet/chet-2500.cisco.com-new-srvtab from 172.68.1.123 (via Ethernet0): !
[OK - 66/1000 bytes]

chet-2500(config)# aaa authentication login default krb5-telnet krb5
chet-2500(config)#
chet-2500#
%SYS-5-CONFIG_I: Configured from console by console
chet-2500# write term
Building configuration...

Current configuration:
!
! Last configuration change at 14:08:32 PDT Mon May 13 1996
!
version 11.2
service udp-small-servers
service tcp-small-servers
!
hostname chet-2500
!
clock timezone PST -8
clock summer-time PDT recurring
aaa new-model
aaa authentication login default krb5-telnet krb5
aaa authentication login console none
aaa authentication ppp local local
enable password sMudgKin
!
username chet-2500 password 7 sMudgkin
username chet-3000 password 7 sMudgkin
username chetin password 7 sMudgkin
kerberos local-realm CISCO.COM
kerberos srvtab entry host/chet-2500.cisco.com@CISCO.COM 0 832015393 1 1 8 7 sMudgkin
kerberos server CISCO.COM 172.71.54.14
kerberos credentials forward
!
interface Ethernet0
 ip address 172.16.0.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
 no fair-queue
!

interface Serial1
 no ip address
 shutdown
 no fair-queue
!
interface Async2
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
interface Async3
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic address
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
router eigrp 109
 network 172.17.0.0
 no auto-summary
!
ip default-gateway 172.30.55.64
ip domain-name cisco.com
ip name-server 192.168.0.0
ip classless
!
!
line con 0
 exec-timeout 0 0
 login authentication console
line 1 16
 transport input all
line aux 0
 transport input all
line vty 0 4
 password sMudgKin
!
ntp clock-period 17179703
ntp peer 172.19.10.0
ntp peer 172.19.0.0
end

chet-2500#
```

With this configuration, the user can Telnet in to the router using Kerberos credentials, as illustrated in the next example:

```
chet-ss20% bin/telnet -a -F chet-2500
Trying 172.16.0.0...
Connected to chet-2500.cisco.com.
Escape character is '^]'.
[ Kerberos V5 accepts you as "chet@CISCO.COM" ]



User Access Verification

chet-2500>[ Kerberos V5 accepted forwarded credentials ]

chet-2500> show kerberos creds
Default Principal:  chet@CISCO.COM
Valid Starting         Expires                Service Principal
13-May-1996 15:06:25   14-May-1996 00:08:29   krbtgt/CISCO.COM@CISCO.COM

chet-2500>q
Connection closed by foreign host.
chet-ss20%
```

The following example shows how to map Kerberos instances to Cisco's privilege levels. To map Kerberos instances to privilege levels, you would perform the following tasks:

- Entering configuration mode
- Mapping the Kerberos instance admin to privilege level 15
- Mapping the Kerberos instance restricted to privilege level 3
- Specifying that the instance defined by the **kerberos instance map** command be used for AAA Authorization
- Writing the configuration to the terminal

```
chet-2500# configure term
Enter configuration commands, one per line.  End with CNTL/Z.
chet-2500(config)# kerberos instance map admin 15
chet-2500(config)# kerberos instance map restricted 3
chet-2500(config)# aaa authorization exec default krb5-instance
chet-2500(config)#
chet-2500#
%SYS-5-CONFIG_I: Configured from console by console
chet-2500# write term
Building configuration...

Current configuration:
!
! Last configuration change at 14:59:05 PDT Mon May 13 1996
!
version 11.2
service udp-small-servers
service tcp-small-servers
!
hostname chet-2500
!
aaa new-model
aaa authentication login default krb5-telnet krb5
aaa authentication login console none
aaa authentication ppp default krb5 local
aaa authorization exec default krb5-instance
enable password sMudgKin
```

```
!
username chet-2500 password 7 sMudgkin
username chet-3000 password 7 sMudgkin
username chetin password 7 sMudgkin
ip domain-name cisco.com
ip name-server 192.168.0.0
kerberos local-realm CISCO.COM
kerberos srvtab entry host/chet-2500.cisco.com@CISCO.COM 0 832015393 1 1 8 7 sMudgkin
kerberos server CISCO.COM 172.71.54.14
kerberos instance map admin 15
kerberos instance map restricted 3
kerberos credentials forward
clock timezone PST -8
clock summer-time PDT recurring
!
interface Ethernet0
 ip address 172.16.0.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
 no fair-queue
!
interface Serial1
 no ip address
 shutdown
 no fair-queue
!
interface Async2
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
interface Async3
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic address
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
router eigrp 109
 network 172.17.0.0
no auto-summary
!
ip default-gateway 172.30.55.64
ip classless
!
!
line con 0
 exec-timeout 0 0
 login authentication console
line 1 16
 transport input all
line aux 0
 transport input all
line vty 0 4
```

```
 password sMudgKin
!
ntp clock-period 17179703
ntp peer 172.19.10.0
ntp peer 172.19.0.0
end

chet-2500#
```

The following example shows output from the three types of sessions now possible for user chet with Kerberos instances turned on:

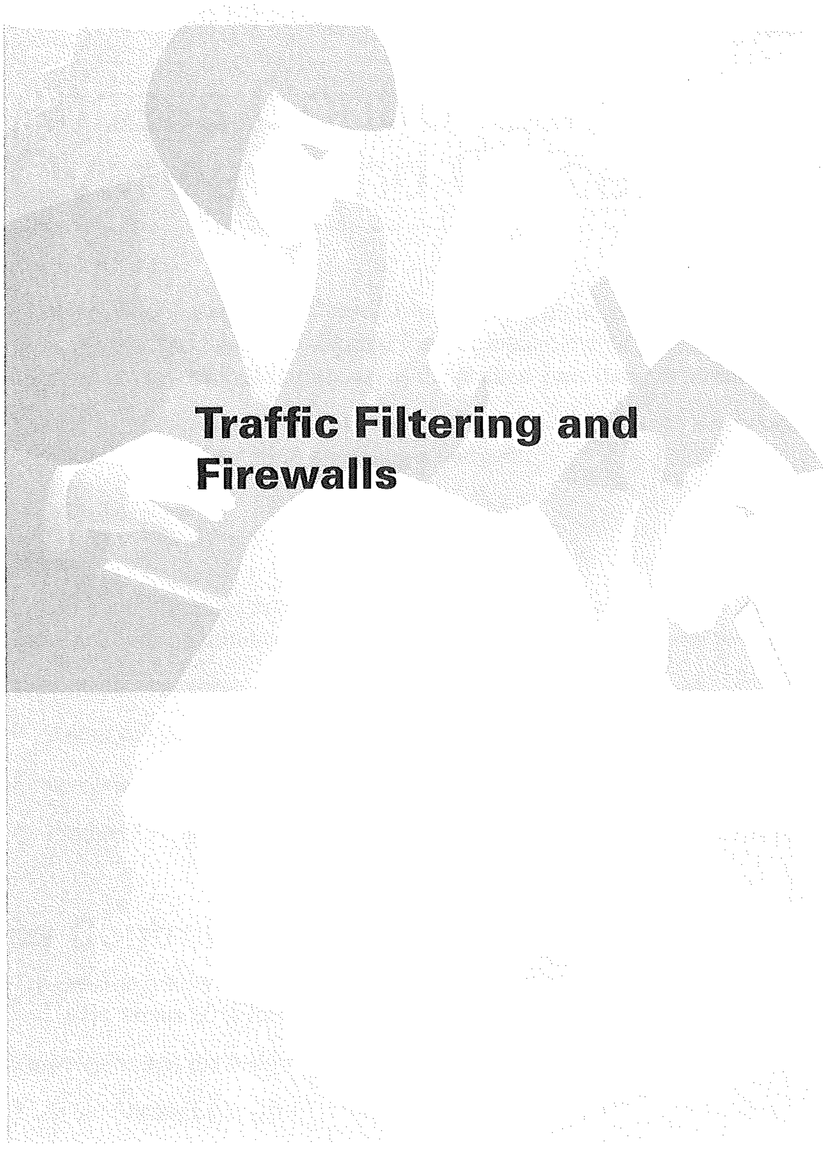
```
chet-ss20% telnet chet-2500
Trying 172.16.0.0 ...
Connected to chet-2500.cisco.com.
Escape character is '^]'.

User Access Verification

Username: chet
Password:

chet-2500> show kerberos creds
Default Principal:  chet@CISCO.COM
Valid Starting          Expires                 Service Principal
13-May-1996 14:58:28    13-May-1996 22:59:29    krbtgt/CISCO.COM@CISCO.COM

chet-2500> show privilege
Current privilege level is 1
chet-2500> q
Connection closed by foreign host.
chet-ss20% telnet chet-2500
Trying 172.16.0.0 ...
Connected to chet-2500.cisco.com.
Escape character is '^]'.


User Access Verification

Username: chet/admin
Password:

chet-2500# show kerberos creds
Default Principal:  chet/admin@CISCO.COM
Valid Starting          Expires                 Service Principal
13-May-1996 14:59:44    13-May-1996 23:00:45    krbtgt/CISCO.COM@CISCO.COM
chet-2500# show privilege
Current privilege level is 15
chet-2500# q
Connection closed by foreign host.
chet-ss20% telnet chet-2500
Trying 172.16.0.0 ...
Connected to chet-2500.cisco.com.
Escape character is '^]'.

User Access Verification

Username: chet/restricted
Password:

chet-2500# show kerberos creds
Default Principal:  chet/restricted@CISCO.COM
Valid Starting          Expires                 Service Principal
13-May-1996 15:00:32    13-May-1996 23:01:33    krbtgt/CISCO.COM@CISCO.COM
```

```
chet-2500# show privilege
Current privilege level is 3
chet-2500# q
Connection closed by foreign host.
chet-ss20%
```

Encrypted Telnet Session Example

The following example shows how to establish an encrypted Telnet session from a router to a remote host named "host1":

```
Router> telnet host1 /encrypt kerberos
```



Traffic Filtering and Firewalls

Access Control Lists: Overview and Guidelines

Cisco provides basic traffic filtering capabilities with access control lists (also referred to as *access lists*). Access lists can be configured for all routed network protocols (IP, AppleTalk, and so on) to filter the packets of those protocols as the packets pass through a router.

You can configure access lists at your router to control access to a network: access lists can prevent certain traffic from entering or exiting a network.

In This Chapter

This chapter describes access lists as part of a security solution. This chapter includes tips, cautions, considerations, recommendations, and general guidelines for how to use access lists.

This chapter has these sections:

- About Access Control Lists
- Overview of Access List Configuration
- Finding Complete Configuration and Command Information for Access Lists

About Access Control Lists

This section briefly describes what access lists do; why and when you should configure access lists; and basic versus advanced access lists.

This section has the following sections:

- What Access Lists Do
- Why You Should Configure Access Lists
- When to Configure Access Lists
- Basic Vs. Advanced Access Lists

What Access Lists Do

Access lists filter network traffic by controlling whether routed packets are forwarded or blocked at the router's interfaces. Your router examines each packet to determine whether to forward or drop the packet, based on the criteria you specified within the access lists.

Access list criteria could be the source address of the traffic, the destination address of the traffic, the upper-layer protocol, or other information. Note that sophisticated users can sometimes successfully evade or fool basic access lists because no authentication is required.

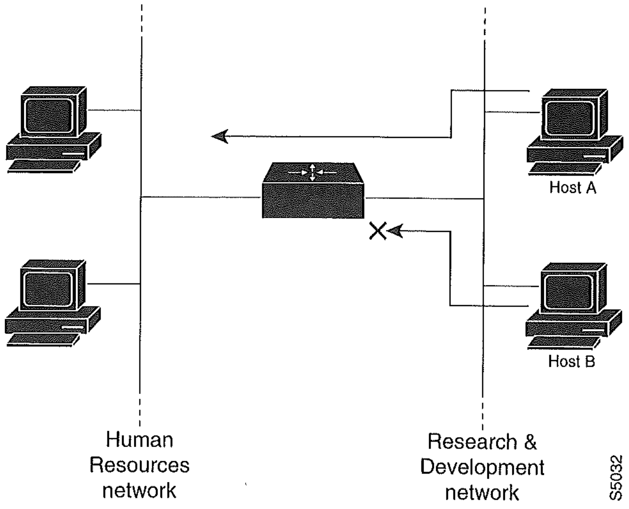
Why You Should Configure Access Lists

There are many reasons to configure access lists—for example, you can use access lists to restrict contents of routing updates, or to provide traffic flow control. But one of the most important reasons to configure access lists is to provide security for your network, which is the focus of this chapter.

You should use access lists to provide a basic level of security for accessing your network. If you do not configure access lists on your router, all packets passing through the router could be allowed onto all parts of your network.

For example, access lists can allow one host to access a part of your network, and prevent another host from accessing the same area. In Figure 9, Host A is allowed to access the Human Resources network and Host B is prevented from accessing the Human Resources network.

Figure 9 Using Traffic Filters to Prevent Traffic from Being Routed to a Network



You can also use access lists to decide which types of traffic are forwarded or blocked at the router interfaces. For example, you can permit e-mail traffic to be routed, but at the same time block all Telnet traffic.

When to Configure Access Lists

Access lists should be used in "firewall" routers, which are often positioned between your internal network and an external network such as the Internet. You can also use access lists on a router positioned between two parts of your network, to control traffic entering or exiting a specific part of your internal network.

To provide the security benefits of access lists, you should at a minimum configure access lists on border routers—routers situated at the edges of your networks. This provides a basic buffer from the outside network, or from a less controlled area of your own network into a more sensitive area of your network.

On these routers, you should configure access lists for each network protocol configured on the router interfaces. You can configure access lists so that inbound traffic or outbound traffic or both are filtered on an interface.

Access lists must be defined on a per-protocol basis. In other words, you should define access lists for every protocol enabled on an interface if you want to control traffic flow for that protocol.

Note Some protocols refer to access lists as *filters*.

Basic Vs. Advanced Access Lists

This chapter describes how to use standard and static extended access lists, which are the basic types of access lists. Some type of basic access list should be used with each routed protocol that you have configured for router interfaces.

Besides the basic types of access lists described in this chapter, there are also more advanced access lists available, which provide additional security features and give you greater control over packet transmission. These advanced access lists and features are described in the other chapters within the "Traffic Filtering and Firewalls" section.

Overview of Access List Configuration

Although each protocol has its own set of specific tasks and rules required for you to provide traffic filtering, in general most protocols require at least two basic steps to be accomplished. The first step is to create an access list definition, and the second step is to apply the access list to an interface.

The following sections describe these two steps:

- Creating Access Lists
- Applying Access Lists to Interfaces

Note that some protocols refer to access lists as *filters* and refer to the act of applying the access lists to interfaces as *filtering*.

Creating Access Lists

Create access lists for each protocol you wish to filter, per router interface. For some protocols, you create one access list to filter inbound traffic, and one access list to filter outbound traffic.

To create an access list, you specify the protocol to filter, you assign a unique name or number to the access list, and you define packet filtering criteria. A single access list can have multiple filtering criteria statements.

Cisco recommends that you create your access lists on a TFTP server, then download the access lists to your router. This can considerably simplify maintenance of your access lists. For details, see the "Creating and Editing Access List Statements on a TFTP Server" section later in this chapter.

The protocols for which you can configure access lists are identified in Table 14 and Table 15.

This section has the following sections:

- Assigning a Unique Name or Number to Each Access List
- Defining Criteria for Forwarding or Blocking Packets
- Creating and Editing Access List Statements on a TFTP Server

Assigning a Unique Name or Number to Each Access List

When configuring access lists on a router, you must identify each access list uniquely within a protocol, by assigning either a name or a number to the protocol's access list.

Note Access lists of some protocols must be identified by a name, and access lists of other protocols must be identified by a number. Some protocols can be identified by either a name or a number. When a number is used to identify an access list, the number must be within the specific range of numbers that is valid for the protocol.

You can specify access lists by names for the protocols listed in Table 14.

Table 14 Protocols with Access Lists Specified by Names

Protocol
Apollo Domain
IP
IPX
ISO CLNS
NetBIOS IPX
Source-route bridging NetBIOS

You can specify access lists by numbers for the protocols listed in Table 15. Table 15 also lists the range of access list numbers that is valid for each protocol.

Table 15 Protocols with Access Lists Specified by Numbers

Protocol	Range
IP	1 to 99 and 1300 to 1999
Extended IP	100 to 199 and 2000 to 2699
Ethernet type code	200 to 299
Ethernet address	700 to 799
Transparent bridging (protocol type)	200 to 299
Transparent bridging (vendor code)	700 to 799
Extended transparent bridging	1100 to 1199
DECnet and extended DECnet	300 to 399
XNS	400 to 499
Extended XNS	500 to 599
AppleTalk	600 to 699
Source-route bridging (protocol type)	200 to 299
Source-route bridging (vendor code)	700 to 799
IPX	800 to 899
Extended IPX	900 to 999
IPX SAP	1000 to 1099

Table 15 **Protocols with Access Lists Specified by Numbers (continued)**

Protocol	Range
Standard VINES	1 to 100
Extended VINES	101 to 200
Simple VINES	201 to 300

Defining Criteria for Forwarding or Blocking Packets

When creating an access list, you define criteria which are applied to each packet that is processed by the router; the router decides whether to forward or block each packet based on whether or not the packet matches the criteria.

Typical criteria you define in access lists are packet source addresses, packet destination addresses, or upper-layer protocol of the packet. However, each protocol has its own specific set of criteria that can be defined.

For a single access list, you can define multiple criteria in multiple, separate access list statements. Each of these statements should reference the same identifying name or number, to tie the statements to the same access list. You can have as many criteria statements as you want, limited only by the available memory. Of course, the more statements you have, the more difficult it will be to comprehend and manage your access lists.

The Implied "Deny All Traffic" Criteria Statement

At the end of every access list is an implied "deny all traffic" criteria statement. Therefore, if a packet does not match any of your criteria statements, the packet will be blocked.

Note For most protocols, if you define an inbound access list for traffic filtering, you should include explicit access list criteria statements to permit routing updates. If you do not, you might effectively lose communication from the interface when routing updates are blocked by the implicit "deny all traffic" statement at the end of the access list.

The Order in Which You Enter Criteria Statements

Note that each additional criteria statement that you enter is appended to the *end* of the access list statements. Also note that you cannot delete individual statements after they have been created. You can only delete an entire access list.

The order of access list statements is important! When the router is deciding whether to forward or block a packet, the Cisco IOS software tests the packet against each criteria statement in the order the statements were created. After a match is found, no more criteria statements are checked.

If you create a criteria statement that explicitly permits all traffic, no statements added later will ever be checked. If you need additional statements, you must delete the access list and retype it with the new entries.

Creating and Editing Access List Statements on a TFTP Server

Because the order of access list criteria statements is important, and because you cannot reorder or delete criteria statements on your router, Cisco recommends that you create all access list statements on a TFTP server, and then download the entire access list to your router.

To use a TFTP server, create the access list statements using any text editor, and save the access list in ASCII format to a TFTP server that is accessible by your router. Then, from your router, use the **copy tftp:***file_id* **system:running-config** command to copy the access list to your router. Finally, perform the **copy system:running-config nvram:startup-config** command to save the access list to your router's NVRAM.

Then, if you ever want to make changes to an access list, you can make them to the text file on the TFTP server, and copy the edited file to your router as before.

Note The first command of an edited access list file should delete the previous access list (for example, type a **no access-list** command at the beginning of the file). If you do not first delete the previous version of the access list, when you copy the edited file to your router you will merely be appending additional criteria statements to the end of the existing access list.

Applying Access Lists to Interfaces

For some protocols, you can apply up to two access lists to an interface: one inbound access list and one outbound access list. With other protocols, you apply only one access list which checks both inbound and outbound packets.

If the access list is inbound, when the router receives a packet, the Cisco IOS software checks the access list's criteria statements for a match. If the packet is permitted, the software continues to process the packet. If the packet is denied, the software discards the packet.

If the access list is outbound, after receiving and routing a packet to the outbound interface, the software checks the access list's criteria statements for a match. If the packet is permitted, the software transmits the packet. If the packet is denied, the software discards the packet.

Finding Complete Configuration and Command Information for Access Lists

The guidelines discussed in this chapter apply in general to all protocols. The specific instructions for creating access lists and applying them to interfaces vary from protocol to protocol, and this specific information is not included in this chapter.

To find complete configuration and command information to configure access lists for a specific protocol, see the appropriate protocol's chapters in the Cisco IOS configuration guides and command references. For example, to configure access lists for the IP protocol, refer to the "Configuring IP Services" chapter in the *Cisco IOS IP and IP Routing Configuration Guide.*

For information on dynamic access lists, see the chapter "Configuring Lock-and-Key Security (Dynamic Access Lists)" in this guide.

For information on reflexive access lists, see the chapter "Configuring IP Session Filtering (Reflexive Access Lists)" in this guide.

Cisco Secure Integrated Software Firewall Overview

This chapter describes how you can configure your Cisco networking device to function as a firewall, using Cisco Secure Integrated Software security features.

This chapter has the following sections:

- Overview of Firewalls
- The Cisco Secure Integrated Software Firewall Solution
- Creating a Customized Firewall
- Other Guidelines for Configuring Your Firewall

Overview of Firewalls

Firewalls are networking devices that control access to your organization's network assets. Firewalls are positioned at the entrance points into your network. If your network has multiple entrance points, you must position a firewall at each point to provide effective network access control.

Firewalls are often placed in between the internal network and an external network such as the Internet. With a firewall between your network and the Internet, all traffic coming from the Internet must pass through the firewall before entering your network.

Firewalls can also be used to control access to a specific part of your network. For example, you can position firewalls at all the entry points into a research and development network to prevent unauthorized access to proprietary information.

The most basic function of a firewall is to monitor and filter traffic. Firewalls can be simple or elaborate, depending on your network requirements. Simple firewalls are usually easier to configure and manage. However, you might require the flexibility of a more elaborate firewall.

The Cisco Secure Integrated Software Firewall Solution

Cisco IOS software provides an extensive set of security features, allowing you to configure a simple or elaborate firewall, according to your particular requirements. You can configure a Cisco device as a firewall if the device is positioned appropriately at a network entry point. Security features that provide firewall functionality are listed in the "Creating a Customized Firewall" section.

In addition to the security features available in standard Cisco IOS feature sets, Cisco Secure Integrated Software gives your router additional firewall capabilities.

The Cisco Secure Integrated Software Feature Set

The Cisco Secure Integrated Software (IS) feature set combines existing Cisco IOS firewall technology and the Context-based Access Control (CBAC) feature. When you configure the Cisco Secure IS on your Cisco router, you turn your router into an effective, robust firewall.

The Cisco Secure IS features are designed to prevent unauthorized, external individuals from gaining access to your internal network, and to block attacks on your network, while at the same time allowing authorized users to access network resources.

You can use the Cisco Secure IS features to configure your Cisco IOS router as follows:

- An Internet firewall or part of an Internet firewall
- A firewall between groups in your internal network
- A firewall providing secure connections to or from branch offices
- A firewall between your company's network and your company's partners' networks

The Cisco Secure IS features provide the following benefits:

- Protection of internal networks from intrusion
- Monitoring of traffic through network perimeters
- Enabling of network commerce via the World Wide Web

Creating a Customized Firewall

To create a firewall customized to fit your organization's security policy, you should determine which Cisco Secure IS features are appropriate, and configure those features. At a minimum, you must configure basic traffic filtering to provide a basic firewall. You can configure your Cisco networking device to function as a firewall by using the following Cisco Secure IS features:

- Standard Access Lists and Static Extended Access Lists
- Lock-and-Key (Dynamic Access Lists)
- Reflexive Access Lists
- TCP Intercept
- Context-based Access Control
- Cisco Secure Integrated Software Intrusion Detection System
- Authentication Proxy
- Port to Application Mapping
- Security Server Support
- Network Address Translation
- IPSec Network Security
- Neighbor Router Authentication
- Event Logging
- User Authentication and Authorization

As well as configuring these features, you should follow the guidelines listed in the "Other Guidelines for Configuring Your Firewall" section. This section outlines important security practices to protect your firewall and network. Table 16 describes Cisco IOS security features.

Table 16 Cisco IOS Features for a Robust Firewall

Feature	Chapter	Comments
Standard Access Lists and Static Extended Access Lists	"Access Control Lists: Overview and Guidelines"	Standard and static extended access lists provide basic traffic filtering capabilities. You configure criteria that describe which packets should be forwarded, and which packets should be dropped at an interface, based on each packet's network layer information. For example, you can block all 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 block or forward packets. To configure a basic firewall, you should at a minimum configure basic traffic filtering. You should configure basic access lists for all network protocols that will be routed through your firewall, such as IP, IPX, AppleTalk, and so forth.
Lock-and-Key (Dynamic Access Lists)	"Configuring Lock-and-Key Security (Dynamic Access Lists)"	Lock-and-Key provides 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) before the firewall allows their traffic through the firewall. Afterwards, the firewall closes the temporary opening. This provides tighter control over traffic at the firewall than with standard or static extended access lists.
Reflexive Access Lists	"Configuring IP Session Filtering (Reflexive Access Lists)"	Reflexive access lists filter IP traffic so that TCP or UDP "session" traffic is only permitted through the firewall if the session originated from within the internal network. You would only configure Reflexive Access Lists when not using Context-based Access Control.
TCP Intercept	"Configuring TCP Intercept (Prevent Denial-of-Service Attacks)"	TCP Intercept protects TCP servers within your network from TCP SYN-flooding attacks, a type of denial-of-service attack. You would only configure TCP Intercept when not using Context-based Access Control.
Context-based Access Control	"Configuring Context-Based Access Control"	Context-based Access Control (CBAC) examines not only network layer and transport layer information, but also examines the application-layer protocol information (such as FTP information) to learn about the state of TCP and UDP connections. CBAC maintains connection state information for individual connections. This state information is used to make intelligent decisions about whether packets should be permitted or denied, and dynamically creates and deletes temporary openings in the firewall. CBAC also generates real-time alerts and audit trails. Enhanced audit trail features use SYSLOG to track all network transactions. Real-time alerts send SYSLOG error messages to central management consoles upon detecting suspicious activity. Using CBAC inspection rules, you can configure alerts and audit trail information on a per-application protocol basis. CBAC is only available in the Cisco Secure Integrated Software feature set.

Table 16 **Cisco IOS Features for a Robust Firewall (continued)**

Feature	Chapter	Comments
Cisco Secure Integrated Software Intrusion Detection System	"Configuring Cisco Secure Integrated Software Intrusion Detection System"	The Cisco Secure Integrated Software Intrusion Detection System (IDS) acts as an in-line intrusion detection sensor, watching packets and sessions as they flow through the router, scanning each to match any of the IDS signatures. When it detects suspicious activity, it responds before network security can be compromised and logs the event through Cisco IOS syslog. The network administrator can configure the IDS system to choose the appropriate response to various threats. When packets in a session match a signature, the IDS system can be configured to: • Send an alarm to a syslog server or a Cisco NetRanger Director (centralized management interface) • Drop the packet • Reset the TCP connection
Authentication Proxy	"Configuring Authentication Proxy"	The Cisco Secure Integrated Software authentication proxy feature allows network administrators to apply specific security policies on a per-user basis. Previously, user identity and related authorized access was associated with a user's IP address, or a single security policy had to be applied to an entire user group or sub network. Now, users can be identified and authorized on the basis of their per-user policy, and access privileges tailored oñ an individual basis are possible, as opposed to general policy applied across multiple users.
Port to Application Mapping	"Configuring Port to Application Mapping"	Port to Application Mapping (PAM) is a feature of Cisco Secure Integrated Software. PAM allows you to customize TCP or UDP port numbers for network services or applications. PAM uses this information to support network environments that run services using ports that are different from the registered or well-known ports associated with an application. The information in the PAM table enables CBAC supported services to run on non-standard ports.
Security Server Support	"Configuring TACACS+," "Configuring RADIUS," and "Configuring Kerberos"	The Cisco Secure Integrated Software feature set can be configured as a client of the following supported security servers: • TACACS+ (including CiscoSecure) • RADIUS • Kerberos You can use any of these security servers to store a database of user profiles. To gain access into your firewall or to gain access through the firewall into another network, users must enter authentication information (such as a username and password), which is matched against the information on the security server. When users pass authentication, they are granted access according to their specified privileges.

Table 16 Cisco IOS Features for a Robust Firewall (continued)

Feature	Chapter	Comments
Network Address Translation	"Configuring IP Addressing" chapter in the *Cisco IOS IP and IP Routing Configuration Guide*	You can use Network Address Translation (NAT) to hide internal IP network addresses from the world outside the firewall. NAT was designed to provide IP address conservation and for internal IP networks that have unregistered (not globally unique) IP addresses: NAT translates these unregistered IP addresses into legal addresses at the firewall. NAT can also be configured to advertise only one address for the entire internal network to the outside world. This provides security by effectively hiding the entire internal network from the world. NAT gives you limited spoof protection because internal addresses are hidden. Additionally, NAT removes all your internal services from the external name space. NAT does not work with the application-layer protocols RPC, VDOLive, or SQL*Net "Redirected." (NAT does work with SQL*Net "Bequeathed.") Do not configure NAT with networks that will carry traffic for these incompatible protocols.
IPSec Network Security	"Configuring IPSec Network Security"	IPSec is a framework of open standards developed by the Internet Engineering Task Force (IETF) that provides security for transmission of sensitive information over unprotected networks such as the Internet. IPSec acts at the network layer, protecting and authenticating IP packets between participating IPSec devices ("peers") such as Cisco routers.
Neighbor Router Authentication	"Neighbor Router Authentication: Overview and Guidelines"	Neighbor router authentication requires the firewall to authenticate all neighbor routers before accepting any route updates from that neighbor. This ensures that the firewall receives legitimate route updates from a trusted source.
Event Logging	"Troubleshooting the Router" chapter in the "System Management" part of the *Cisco IOS Configuration Fundamentals Configuration Guide*	Event logging automatically logs output from system error messages and other events to the console terminal. You can also redirect these messages to other destinations such as virtual terminals, internal buffers, or syslog servers. You can also specify the severity of the event to be logged, and you can configure the logged output to be timestamped. The logged output can be used to assist real-time debugging and management, and to track potential security breaches or other nonstandard activities throughout a network.
User Authentication and Authorization	"Configuring Authentication" and "Configuring Authorization"	Authentication and authorization help protect your network from access by unauthorized users.

Other Guidelines for Configuring Your Firewall

As with all networking devices, you should always protect access into the firewall by configuring passwords as described in the chapter "Configuring Passwords and Privileges." You should also consider configuring user authentication, authorization, and accounting as described in the chapters in the "Authentication, Authorization, and Accounting (AAA)" part of this guide.

You should also consider the following recommendations:

- When setting passwords for privileged access to the firewall, use the **enable secret** command rather than the **enable password** command, which does not have as strong an encryption algorithm.
- Put a password on the console port. In authentication, authorization, and accounting (AAA) environments, use the same authentication for the console as for elsewhere. In a non-AAA environment, at a minimum configure the **login** and **password** *password* commands.
- Think about access control *before* you connect a console port to the network in any way, including attaching a modem to the port. Be aware that a *break* on the console port might give total control of the firewall, even with access control configured.
- Apply access lists and password protection to all virtual terminal ports. Use access lists to limit who can Telnet into your router.
- Do not enable any local service (such as SNMP or NTP) that you do not use. Cisco Discovery Protocol (CDP) and Network Time Protocol (NTP) are on by default, and you should turn these off if you do not need them.

  To turn off CDP, enter the **no cdp run** global configuration command. To turn off NTP, enter the **ntp disable** interface configuration command on each interface not using NTP.

  If you must run NTP, configure NTP only on required interfaces, and configure NTP to listen only to certain peers.

  Any enabled service could present a potential security risk. A determined, hostile party might be able to find creative ways to misuse the enabled services to access the firewall or the network.

  For local services that are enabled, protect against misuse. Protect by configuring the services to communicate only with specific peers, and protect by configuring access lists to deny packets for the services at specific interfaces.
- Protect against spoofing: protect the networks on both sides of the firewall from being spoofed from the other side. You could protect against spoofing by configuring input access lists at all interfaces to pass only traffic from expected source addresses, and to deny all other traffic.

  You should also disable source routing. For IP, enter the **no ip source-route** global configuration command. Disabling source routing at *all* routers can also help prevent spoofing.

  You should also disable minor services. For IP, enter the **no service tcp-small-servers** and **no service udp-small-servers** global configuration commands.
- Prevent the firewall from being used as a relay by configuring access lists on any asynchronous Telnet ports.
- Normally, you should disable directed broadcasts for all applicable protocols on your firewall and on all your other routers. For IP, use the **no ip directed-broadcast** command. Rarely, some IP networks do require directed broadcasts; if this is the case, do not disable directed broadcasts.

  Directed broadcasts can be misused to multiply the power of denial-of-service attacks, because every denial-of-service packet sent is broadcast to every host on a subnet. Furthermore, some hosts have other intrinsic security risks present when handling broadcasts.
- Configure the **no proxy-arp** command to prevent internal addresses from being revealed. (This is important to do if you do not already have NAT configured to prevent internal addresses from being revealed).
- Keep the firewall in a secured (locked) room.

Configuring Lock-and-Key Security (Dynamic Access Lists)

This chapter describes how to configure lock-and-key security at your router. Lock-and-key is a traffic filtering security feature available for the IP protocol.

For a complete description of lock-and-key commands, refer to the "Lock-and-Key Commands" chapter of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

In This Chapter

This chapter has the following sections:

- About Lock-and-Key
- Compatibility with Releases Before Cisco IOS Release 11.1
- Risk of Spoofing with Lock-and-Key
- Router Performance Impacts with Lock-and-Key
- Prerequisites to Configuring Lock-and-Key
- Configuring Lock-and-Key
- Verifying Lock-and-Key Configuration
- Maintaining Lock-and-Key
- Lock-and-Key Configuration Examples

About Lock-and-Key

Lock-and-key is a traffic filtering security feature that dynamically filters IP protocol traffic. Lock-and-key is configured using IP dynamic extended access lists. Lock-and-key can be used in conjunction with other standard access lists and static extended access lists.

When lock-and-key is configured, designated users whose IP traffic is normally blocked at a router can gain temporary access through the router. When triggered, lock-and-key reconfigures the interface's existing IP access list to permit designated users to reach their designated host(s). Afterwards, lock-and-key reconfigures the interface back to its original state.

For a user to gain access to a host through a router with lock-and-key configured, the user must first Telnet to the router. When a user initiates a standard Telnet session to the router, lock-and-key automatically attempts to authenticate the user. If the user is authenticated, they will then gain temporary access through the router and be able to reach their destination host.

This section has the following sections:

- Benefits of Lock-and-Key
- When to Use Lock-and-Key
- How Lock-and-Key Works

Benefits of Lock-and-Key

Lock-and-key provides the same benefits as standard and static extended access lists (these benefits are discussed in the chapter "Access Control Lists: Overview and Guidelines"). However, lock-and-key also has the following security benefits over standard and static extended access lists:

- Lock-and-key uses a challenge mechanism to authenticate individual users.
- Lock-and-key provides simpler management in large internetworks.
- In many cases, lock-and-key reduces the amount of router processing required for access lists.
- Lock-and-key reduces the opportunity for network break-ins by network hackers.

With lock-and-key, you can specify which users are permitted access to which source/destination hosts. These users must pass a user authentication process before they are permitted access to their designated host(s). Lock-and-key creates dynamic user access through a firewall, without compromising other configured security restrictions.

When to Use Lock-and-Key

Two examples of when you might use lock-and-key follow:

- When you want a specific remote user (or group of remote users) to be able to access a host within your network, connecting from their remote host(s) via the Internet. Lock-and-key authenticates the user, then permits limited access through your firewall router for the individual's host or subnet, for a finite period of time.
- When you want a subset of hosts on a local network to access a host on a remote network protected by a firewall. With lock-and-key, you can enable access to the remote host only for the desired set of local user's hosts. Lock-and-key require the users to authenticate through a TACACS+ server, or other security server, before allowing their hosts to access the remote hosts.

How Lock-and-Key Works

The following process describes the lock-and-key access operation:

1. A user opens a Telnet session to a border (firewall) router configured for lock-and-key. The user connects via the virtual terminal port on the router.
2. The Cisco IOS software receives the Telnet packet, opens a Telnet session, prompts for a password, and performs a user authentication process. The user must pass authentication before access through the router is allowed. The authentication process can be done by the router or by a central access security server such as a TACACS+ or RADIUS server.
3. When the user passes authentication, they are logged out of the Telnet session, and the software creates a temporary entry in the dynamic access list. (Per your configuration, this temporary entry can limit the range of networks to which the user is given temporary access.)

4 The user exchanges data through the firewall.

5 The software deletes the temporary access list entry when a configured timeout is reached, or when the system administrator manually clears it. The configured timeout can either be an idle timeout or an absolute timeout.

Note The temporary access list entry is not automatically deleted when the user terminates a session. The temporary access list entry remains until a configured timeout is reached or until it is cleared by the system administrator.

Compatibility with Releases Before Cisco IOS Release 11.1

Enhancements to the **access-list** command are used for lock-and-key. These enhancements are backward compatible—if you migrate from a release before Cisco IOS Release 11.1 to a newer release, your access lists will be automatically converted to reflect the enhancements. However, if you try to use lock-and-key with a release before Cisco IOS Release 11.1, you might encounter problems as described in the following caution paragraph:

Caution Cisco IOS releases before Release 11.1 are not upwardly compatible with the lock-and-key access list enhancements. Therefore, if you save an access list with software older than Release 11.1, and then use this software, the resulting access list will not be interpreted correctly. *This could cause you severe security problems.* You must save your old configuration files with Cisco IOS Release 11.1 or later software before booting an image with these files.

Risk of Spoofing with Lock-and-Key

Caution Lock-and-key access allows an external event (a Telnet session) to place an opening in the firewall. While this opening exists, the router is susceptible to source address spoofing.

When lock-and-key is triggered, it creates a dynamic opening in the firewall by temporarily reconfiguring an interface to allow user access. While this opening exists, another host might spoof the authenticated user's address to gain access behind the firewall. Lock-and-key does not cause the address spoofing problem; the problem is only identified here as a concern to the user. Spoofing is a problem inherent to all access lists, and lock-and-key does not specifically address this problem.

To prevent spoofing, you could configure network data encryption as described in the chapter "Configuring Cisco Encryption Technology." Configure encryption so that traffic from the remote host is encrypted at a secured remote router, and decrypted locally at the router interface providing lock-and-key. You want to ensure that all traffic using lock-and-key will be encrypted when entering the router; this way no hackers can spoof the source address, because they will be unable to duplicate the encryption or to be authenticated as is a required part of the encryption setup process.

Router Performance Impacts with Lock-and-Key

When lock-and-key is configured, router performance can be affected in the following ways:

- When lock-and-key is triggered, the dynamic access list forces an access list rebuild on the silicon switching engine (SSE). This causes the SSE switching path to slow down momentarily.
- Dynamic access lists require the idle timeout facility (even if the timeout is left to default) and therefore cannot be SSE switched. These entries must be handled in the protocol fast-switching path.
- When remote users trigger lock-and-key at a border router, additional access list entries are created on the border router interface. The interface's access list will grow and shrink dynamically. Entries are dynamically removed from the list after either the idle-timeout or max-timeout period expires. Large access lists can degrade packet switching performance, so if you notice performance problems, you should look at the border router configuration to see if you should remove temporary access list entries generated by lock-and-key.

Prerequisites to Configuring Lock-and-Key

Lock-and-key uses IP extended access lists. You must have a solid understanding of how access lists are used to filter traffic, before you attempt to configure lock-and-key. Access lists are described in the previous chapter, "Access Control Lists: Overview and Guidelines."

Lock-and-key employs user authentication and authorization as implemented in Cisco's authentication, authorization, and accounting (AAA) paradigm. You must understand how to configure AAA user authentication and authorization before you configure lock-and-key. User authentication and authorization is explained in the "Authentication, Authorization, and Accounting (AAA)" part of this document.

Lock-and-key uses the **autocommand** command, which you should understand. This command is described in the "Modem Support and Asynchronous Device Commands" chapter of the *Cisco IOS Dial Solutions Command Reference.*

Configuring Lock-and-Key

To configure lock-and-key, use the following commands beginning in global configuration mode. While completing these steps, be sure to follow the guidelines listed in the "Lock-and-Key Configuration Tips" section.

Step	Command	Purpose
1	**access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} **telnet** *source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log**]	Configures a dynamic access list, which serves as a template and place holder for temporary access list entries.
2	**interface** *type number*	Configures an interface.
3	**ip access-group** *access-list-number*	In interface configuration mode, applies the access list to the interface.
4	**line VTY** *line-number* [*ending-line-number*]	In global configuration mode, defines one or more virtual terminal (VTY) ports. If you specify multiple VTY ports, they must all be configured identically because the software hunts for available VTY ports on a round-robin basis. If you do not want to configure all your VTY ports for lock-and-key access, you can specify a group of VTY ports for lock-and-key support only.

Step	Command	Purpose

5	`login tacacs`
	or
	`username name password secret`
	or
	`password password`
	`login local`
		Configures user authentication.

6	`autocommand access-enable [host] [timeout minutes]`	Enables the creation of temporary access list entries. If the **host** argument is *not* specified, all hosts on the entire network are allowed to set up a temporary access list entry. The dynamic access list contains the network mask to enable the new network connection.

For an example of a lock-and-key configuration, see the section "Lock-and-Key Configuration Examples" later in this chapter.

Lock-and-Key Configuration Tips

Before you configure lock-and-key, you should understand the tips discussed in the following sections:

- Dynamic Access Lists
- Lock-and-Key Authentication
- The autocommand Command

Dynamic Access Lists

Use the following tips for configuring dynamic access lists:

- Do *not* create more than one dynamic access list for any one access list. The software only refers to the first dynamic access list defined.
- Do *not* assign the same *dynamic-name* to another access list. Doing so instructs the software to reuse the existing list. All named entries must be globally unique within the configuration.
- Assign attributes to the dynamic access list in the same way you assign attributes for a static access list. The temporary access list entries inherit the attributes assigned to this list.
- Configure Telnet as the protocol, so that the user must Telnet into the router to be authenticated, before they can gain access through the router.
- Either define an idle timeout now with the **timeout** keyword in the **access-enable** command in the **autocommand** command, or define an absolute timeout value later with the **access-list** command. You must define either an idle timeout or an absolute timeout—otherwise, the temporary access list entry will remain configured indefinitely on the interface (even after the user has terminated their session) until the entry is removed manually by an administrator. (You could configure both idle and absolute timeouts if you wish.)
- If you configure an idle timeout, the idle timeout value should be equal to the WAN idle timeout value.
- If you configure both idle and absolute timeouts, the idle timeout value must be less than the absolute timeout value.
- The only values replaced in the temporary entry are the source or destination address, depending whether the access list was in the input access list or output access list. All other attributes, such as port, are inherited from the main dynamic access list.

- Each addition to the dynamic list is always put at the beginning of the dynamic list. You cannot specify the order of temporary access list entries.
- Temporary access list entries are never written to NVRAM.
- To manually clear or to display dynamic access lists, refer to the section "Maintaining Lock-and-Key" later in this chapter.

Lock-and-Key Authentication

There are three possible methods to configure an authentication query process. These three methods are described in this section.

Note Cisco recommends that you use the TACACS+ server for your authentication query process. TACACS+ provides authentication, authorization, and accounting services. It also provides protocol support, protocol specification, and a centralized security database. Using a TACACS+ server is described in the next section, "Method 1—Configuring a Security Server."

Method 1—Configuring a Security Server

Use a network access security server such as TACACS+ server. This method requires additional configuration steps on the TACACS+ server but allows for stricter authentication queries and more sophisticated tracking capabilities:

```
Router# login tacacs
```

Method 2—Configuring the **username** Command

Use the **username** command. This method is more effective because authentication is determined on a user basis:

```
Router# username name password password
```

Method 3—Configuring the **password** and **login** Commands

Use the **password** and **login** commands. This method is less effective because the password is configured for the port, not for the user. Therefore, any user who knows the password can authenticate successfully:

```
Router# password password
Router# login local
```

The **autocommand** Command

Use the following tips for configuring the **autocommand** command:

- If you use a TACACS+ server to authenticate the user, you should configure the **autocommand** command on the TACACS+ server as a per-user autocommand. If you use local authentication, use the **autocommand** command on the line.
- Configure all virtual terminal (VTY) ports with the same **autocommand** command. Omitting an **autocommand** command on a VTY port allows a random host to gain EXEC mode access to the router and does not create a temporary access list entry in the dynamic access list.

- If you did not previously define an idle timeout with the **autocommand access-enable** command, you must define an absolute timeout now with the **access-list** command. You must define either an idle timeout or an absolute timeout—otherwise, the temporary access list entry will remain configured indefinitely on the interface (even after the user has terminated their session) until the entry is removed manually by an administrator. (You could configure both idle and absolute timeouts if you wish.)
- If you configure both idle and absolute timeouts, the absolute timeout value must be greater than the idle timeout value.

Verifying Lock-and-Key Configuration

You can verify that lock-and-key is successfully configured on the router by asking a user to test the connection. The user should be at a host that is permitted in the dynamic access list, and the user should have AAA authentication and authorization configured.

To test the connection, the user should Telnet to the router, allow the Telnet session to close, and then attempt to access a host on the other side of the router. This host must be one that is permitted by the dynamic access list. The user should access the host with an application that uses the IP protocol.

The following sample display illustrates what end-users might see if they are successfully authenticated. Notice that the Telnet connection is closed immediately after the password is entered and authenticated. The temporary access list entry is then created, and the host that initiated the Telnet session now has access inside the firewall.

```
Router% telnet corporate
Trying 172.21.52.1 ...
Connected to corporate.example.com.
Escape character is '^]'.
User Access Verification
Password:Connection closed by foreign host.
```

You can then use the **show access-lists** command at the router to view the dynamic access lists, which should include an additional entry permitting the user access through the router.

Maintaining Lock-and-Key

When lock-and-key is in use, dynamic access lists will dynamically grow and shrink as entries are added and deleted. You need to make sure that entries are being deleted in a timely way, because while entries exist, the risk of a spoofing attack is present. Also, the more entries there are, the bigger the router performance impact will be.

If you do not have an idle or absolute timeout configured, entries will remain in the dynamic access list until you manually remove them. If this is the case, make sure that you are extremely vigilant about removing entries.

Displaying Dynamic Access List Entries

You can display temporary access list entries when they are in use. After a temporary access list entry is cleared by you or by the absolute or idle timeout parameter, it can no longer be displayed. The number of matches displayed indicates the number of times the access list entry was hit.

To view dynamic access lists and any temporary access list entries that are currently established, use the following command in privileged EXEC mode:

Command	Purpose
show access-lists [*access-list-number*]	Displays dynamic access lists and temporary access list entries.

Manually Deleting Dynamic Access List Entries

To manually delete a temporary access list entry, use the following command in privileged EXEC mode:

Command	Purpose
clear access-template [*access-list-number* | *name*] [*dynamic-name*] [*source*] [*destination*]	Deletes a dynamic access list.

Lock-and-Key Configuration Examples

The following sections provide lock-and-key configuration examples:

- Lock-and-Key with Local Authentication Example
- Lock-and-Key with TACACS+ Authentication Example

Cisco recommends that you use a TACACS+ server for authentication, as shown in the second example.

Lock-and-Key with Local Authentication Example

This example shows how to configure lock-and-key access, with authentication occurring locally at the router. Lock-and-key is configured on the Ethernet 0 interface:

```
interface ethernet0
 ip address 172.18.23.9 255.255.255.0
 ip access-group 101 in

access-list 101 permit tcp any host 172.18.21.2 eq telnet
access-list 101 dynamic mytestlist timeout 120 permit ip any any

line vty 0
login local
autocommand access-enable timeout 5
```

The first access-list entry allows only Telnet into the router. The second access-list entry is always ignored until lock-and-key is triggered.

After a user Telnets into the router, the router will attempt to authenticate the user. If authentication is successful, the **autocommand** executes and the Telnet session terminates. The **autocommand** creates a temporary inbound access list entry at the Ethernet 0 interface, based on the second access-list entry (mytestlist). This temporary entry will expire after 5 minutes, as specified by the timeout.

Lock-and-Key with TACACS+ Authentication Example

The following example shows how to configure lock-and-key access, with authentication on a TACACS+ server. Lock-and-key access is configured on the BRI0 interface. Four VTY ports are defined with the password "cisco":

```
aaa authentication login default group tacacs+ enable
aaa accounting exec stop-only group tacacs+
aaa accounting network stop-only group tacacs+
enable password ciscotac
!
isdn switch-type basic-dms100
!
interface ethernet0
ip address 172.18.23.9 255.255.255.0
!
interface BRI0
 ip address 172.18.21.1 255.255.255.0
 encapsulation ppp
 dialer idle-timeout 3600
 dialer wait-for-carrier-time 100
 dialer map ip 172.18.21.2 name diana
 dialer-group 1
 isdn spid1 2036333715291
 isdn spid2 2036339371566
 ppp authentication chap
 ip access-group 102 in
!
access-list 102 permit tcp any host 172.18.21.2 eq telnet
access-list 102 dynamic testlist timeout 5 permit ip any any
!
!
ip route 172.18.250.0 255.255.255.0 172.18.21.2
priority-list 1 interface BRI0 high
tacacs-server host 172.18.23.21
tacacs-server host 172.18.23.14
tacacs-server key test1
tftp-server rom alias all
!
dialer-list 1 protocol ip permit
!
line con 0
 password cisco
line aux 0
line VTY 0 4
autocommand access-enable timeout 5
password cisco
!
```

Configuring IP Session Filtering (Reflexive Access Lists)

This chapter describes how to configure reflexive access lists on your router. Reflexive access lists provide the ability to filter network traffic at a router, based on IP upper-layer protocol "session" information.

For a complete description of reflexive access list commands, refer to the "Reflexive Access List Commands" chapter of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

In This Chapter

This chapter has the following sections:

- About Reflexive Access Lists
- Prework: Before You Configure Reflexive Access Lists
- Reflexive Access Lists Configuration Task List
- Reflexive Access List Configuration Examples

About Reflexive Access Lists

Reflexive access lists allow IP packets to be filtered based on upper-layer session information. You can use reflexive access lists to permit IP traffic for sessions originating from within your network but to deny IP traffic for sessions originating from outside your network. This is accomplished by reflexive filtering, a kind of session filtering.

Reflexive access lists can be defined with extended named IP access lists only. You cannot define reflexive access lists with numbered or standard named IP access lists or with other protocol access lists.

You can use reflexive access lists in conjunction with other standard access lists and static extended access lists.

This section has the following sections:

- Benefits of Reflexive Access Lists
- What Is a Reflexive Access List?
- How Reflexive Access Lists Implement Session Filtering
- Where to Configure Reflexive Access Lists

- How Reflexive Access Lists Work
- Restrictions on Using Reflexive Access Lists

Benefits of Reflexive Access Lists

Reflexive access lists are an important part of securing your network against network hackers, and can be included in a firewall defense. Reflexive access lists provide a level of security against spoofing and certain denial-of-service attacks. Reflexive access lists are simple to use, and, compared to basic access lists, provide greater control over which packets enter your network.

What Is a Reflexive Access List?

Reflexive access lists are similar in many ways to other access lists. Reflexive access lists contain condition statements (entries) that define criteria for permitting IP packets. These entries are evaluated in order, and when a match occurs, no more entries are evaluated.

However, reflexive access lists have significant differences from other types of access lists. Reflexive access lists contain only temporary entries; these entries are automatically created when a new IP session begins (for example, with an outbound packet), and the entries are removed when the session ends. Reflexive access lists are not themselves applied directly to an interface, but are "nested" within an extended named IP access list that is applied to the interface. (For more information about this, see the section "Reflexive Access Lists Configuration Task List" later in this chapter.) Also, reflexive access lists do not have the usual implicit "deny all traffic" statement at the end of the list, because of the nesting.

How Reflexive Access Lists Implement Session Filtering

This section compares session filtering with basic access lists to session filtering with reflexive access lists. This section contains the following sections:

- With Basic Access Lists
- With Reflexive Access Lists

With Basic Access Lists

With basic standard and static extended access lists, you can approximate session filtering by using the **established** keyword with the **permit** command. The **established** keyword filters TCP packets based on whether the ACK or RST bits are set. (Set ACK or RST bits indicate that the packet is not the first in the session, and therefore, that the packet belongs to an established session.) This filter criterion would be part of an access list applied permanently to an interface.

With Reflexive Access Lists

Reflexive access lists, however, provide a truer form of session filtering, which is much harder to spoof because more filter criteria must be matched before a packet is permitted through. (For example, source and destination addresses and port numbers are checked, not just ACK and RST bits.) Also, session filtering uses temporary filters which are removed when a session is over. This limits the hacker's attack opportunity to a smaller time window.

Moreover, the previous method of using the **established** keyword was available only for the TCP upper-layer protocol. So, for the other upper-layer protocols (such as UDP, ICMP, and so forth), you would have to either permit all incoming traffic or define all possible permissible source/destination host/port address pairs for each protocol. (Besides being an unmanageable task, this could exhaust NVRAM space.)

Where to Configure Reflexive Access Lists

Configure reflexive access lists on border routers—routers that pass traffic between an internal and external network. Often, these are firewall routers.

Note In this chapter, the words "within your network" and "internal network" refer to a network that is controlled (secured), such as your organization's intranet, or to a part of your organization's internal network that has higher security requirements than another part. "Outside your network" and "external network" refer to a network that is uncontrolled (unsecured) such as the Internet or to a part of your organization's network that is not as highly secured.

How Reflexive Access Lists Work

A reflexive access list is triggered when a new IP upper-layer session (such as TCP or UDP) is initiated from inside your network, with a packet traveling to the external network. When triggered, the reflexive access list generates a new, temporary entry. This entry will permit traffic to enter your network if the traffic is part of the session, but will not permit traffic to enter your network if the traffic is not part of the session.

For example, if an outbound TCP packet is forwarded to outside of your network, and this packet is the first packet of a TCP session, then a new, temporary reflexive access list entry will be created. This entry is added to the reflexive access list, which applies to inbound traffic. The temporary entry has characteristics as described next.

This section contains the following sections:

- Temporary Access List Entry Characteristics
- When the Session Ends

Temporary Access List Entry Characteristics

- The entry is always a **permit** entry.
- The entry specifies the same protocol (TCP) as the original outbound TCP packet.
- The entry specifies the same source and destination addresses as the original outbound TCP packet, except the addresses are swapped.
- The entry specifies the same source and destination port numbers as the original outbound TCP packet, except the port numbers are swapped.

  (This entry characteristic applies only for TCP and UDP packets. Other protocols, such as ICMP and IGMP, do not have port numbers, and other criteria are specified. For example, for ICMP, type numbers are used instead.)
- Inbound TCP traffic will be evaluated against the entry, until the entry expires. If an inbound TCP packet matches the entry, the inbound packet will be forwarded into your network.

- The entry will expire (be removed) after the last packet of the session passes through the interface.
- If no packets belonging to the session are detected for a configurable length of time (the timeout period), the entry will expire.

When the Session Ends

Temporary reflexive access list entries are removed at the end of the session. For TCP sessions, the entry is removed 5 seconds after two set FIN bits are detected, or immediately after matching a TCP packet with the RST bit set. (Two set FIN bits in a session indicate that the session is about to end; the 5-second window allows the session to close gracefully. A set RST bit indicates an abrupt session close.) Or, the temporary entry is removed after no packets of the session have been detected for a configurable length of time (the timeout period).

For UDP and other protocols, the end of the session is determined differently than for TCP. Because other protocols are considered to be connectionless (sessionless) services, there is no session tracking information embedded in packets. Therefore, the end of a session is considered to be when no packets of the session have been detected for a configurable length of time (the timeout period).

Restrictions on Using Reflexive Access Lists

Reflexive access lists do not work with some applications that use port numbers that change during a session. For example, if the port numbers for a return packet are different from the originating packet, the return packet will be denied, even if the packet is actually part of the same session.

The TCP application of FTP is an example of an application with changing port numbers. With reflexive access lists, if you start an FTP request from within your network, the request will not complete. Instead, you must use Passive FTP when originating requests from within your network.

Prework: Before You Configure Reflexive Access Lists

Before you configure reflexive access lists, you must decide whether to configure reflexive access lists on an internal or external interface, as described in the next section, "Choosing an Interface: Internal or External."

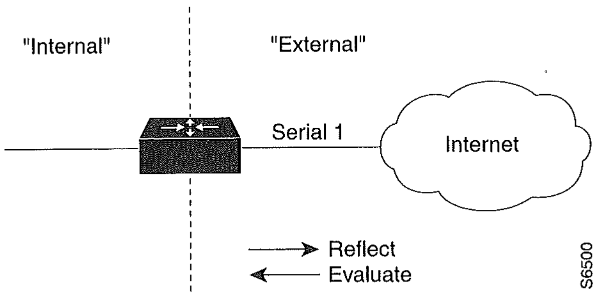
You should also be sure that you have a basic understanding of the IP protocol and of access lists; specifically, you should know how to configure extended named IP access lists. To learn about configuring IP extended access lists, refer to the "Configuring IP Services" chapter of the *Cisco IOS IP and IP Routing Configuration Guide.*

Choosing an Interface: Internal or External

Reflexive access lists are most commonly used with one of two basic network topologies. Determining which of these topologies is most like your own can help you decide whether to use reflexive access lists with an internal interface or with an external interface (the interface connecting to an internal network, or the interface connecting to an external network).

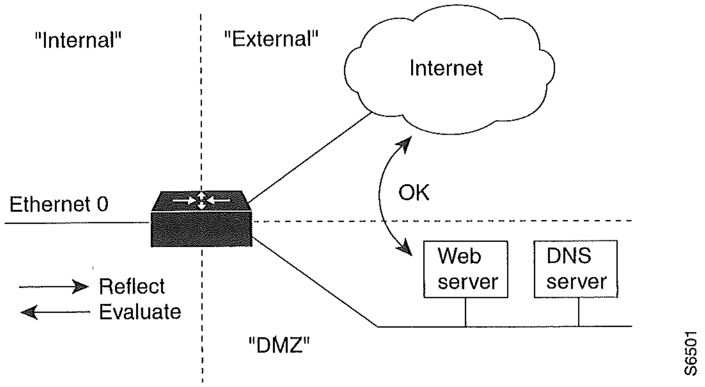
The first topology is shown in Figure 10. In this simple topology, reflexive access lists are configured for the *external* interface Serial 1. This prevents IP traffic from entering the router and the internal network, unless the traffic is part of a session already established from within the internal network.

Figure 10 **Simple Topology—Reflexive Access Lists Configured at the External Interface**



The second topology is shown in Figure 11. In this topology, reflexive access lists are configured for the *internal* interface Ethernet 0. This allows external traffic to access the services in the Demilitarized Zone (DMZ), such as DNS services, but prevents IP traffic from entering your internal network—unless the traffic is part of a session already established from within the internal network.

Figure 11 **DMZ Topology—Reflexive Access Lists Configured at the Internal Interface**



Use these two example topologies to help you decide whether to configure reflexive access lists for an internal or external interface.

Reflexive Access Lists Configuration Task List

In the previous section, "Prework: Before You Configure Reflexive Access Lists," you decided whether to configure reflexive access lists for an internal or external interface.

Now, complete the tasks in one of the following configuration task lists:

- External Interface Configuration Task List
- Internal Interface Configuration Task List

For configuration examples, refer to the "Reflexive Access List Configuration Examples" section at the end of this chapter.

External Interface Configuration Task List

To configure reflexive access lists for an external interface, perform the following tasks:

1. Defining the reflexive access list(s) in an *outbound* IP extended named access list
2. Nesting the reflexive access list(s) in an *inbound* IP extended named access list
3. Setting a global timeout value

These tasks are described in the sections following the "Internal Interface Configuration Task List" section.

Note The defined (outbound) reflexive access list evaluates traffic traveling out of your network: if the defined reflexive access list is matched, temporary entries are created in the nested (inbound) reflexive access list. These temporary entries will then be applied to traffic traveling into your network.

Internal Interface Configuration Task List

To configure reflexive access lists for an internal interface, perform the following tasks:

1. Defining the reflexive access list(s) in an *inbound* IP extended named access list
2. Nesting the reflexive access list(s) in an *outbound* IP extended named access list
3. Setting a global timeout value

These tasks are described in the next sections.

Note The defined (inbound) reflexive access list is used to evaluate traffic traveling out of your network: if the defined reflexive access list is matched, temporary entries are created in the nested (outbound) reflexive access list. These temporary entries will then be applied to traffic traveling into your network.

Defining the Reflexive Access List(s)

To define a reflexive access list, you use an entry in an extended named IP access list. This entry must use the **reflect** keyword.

- If you are configuring reflexive access lists for an external interface, the extended named IP access list should be one that is applied to outbound traffic.
- If you are configuring reflexive access lists for an internal interface, the extended named IP access list should be one that is applied to inbound traffic.

To define reflexive access lists, use the following commands, beginning in global configuration mode:

Step	Command	Purpose
1	`ip access-list extended` *`name`*	External interface: Specifies the outbound access list. or Internal interface: Specifies the inbound access list. (Using this command also causes you to enter the access-list configuration mode.)
2	`permit` *`protocol`* `any any reflect` *`name`* `[timeout` *`seconds`*`]`	Defines the reflexive access list using the reflexive **permit** entry. Repeat this step for each IP upper-layer protocol; for example, you can define reflexive filtering for TCP sessions and also for UDP sessions. You can use the same *name* for multiple protocols. For additional guidelines for this task, see the following section, "Mixing Reflexive Access List Statements with Other Permit and Deny Entries."

If the extended named IP access list you just specified has never been applied to the interface, you must also apply the extended named IP access list to the interface.

To apply the extended named IP access list to the interface, use the following command in interface configuration mode:

Command	Purpose

`ip access-group` *`name`* `out`
or
`ip access-group` *`name`* `in`
	External interface: Applies the extended access list to the interface's outbound traffic.
	Internal interface: Applies the extended access list to the interface's inbound traffic.

Mixing Reflexive Access List Statements with Other Permit and Deny Entries

The extended IP access list that contains the reflexive access list **permit** statement can also contain other normal **permit** and **deny** statements (entries). However, as with all access lists, the order of entries is important, as explained in the next few paragraphs.

If you configure reflexive access lists for an external interface, when an outbound IP packet reaches the interface, the packet will be evaluated sequentially by each entry in the outbound access list until a match occurs.

If the packet matches an entry prior to the reflexive **permit** entry, the packet will not be evaluated by the reflexive **permit** entry, and no temporary entry will be created for the reflexive access list (reflexive filtering will not be triggered).

The outbound packet will be evaluated by the reflexive **permit** entry only if no other match occurs first. Then, if the packet matches the protocol specified in the reflexive **permit** entry, the packet is forwarded out of the interface and a corresponding temporary entry is created in the inbound reflexive access list (unless the corresponding entry already exists, indicating the outbound packet belongs to a session in progress). The temporary entry specifies criteria that permits inbound traffic only for the same session.

Nesting the Reflexive Access List(s)

After you define a reflexive access list in one IP extended access list, you must "nest" the reflexive access list within a different extended named IP access list.

- If you are configuring reflexive access lists for an external interface, nest the reflexive access list within an extended named IP access list applied to inbound traffic.
- If you are configuring reflexive access lists for an internal interface, nest the reflexive access list within an extended named IP access list applied to outbound traffic.

After you nest a reflexive access list, packets heading into your internal network can be evaluated against any reflexive access list temporary entries, along with the other entries in the extended named IP access list.

To nest reflexive access lists, use the following commands, beginning in global configuration mode:

Step	Command	Purpose
1	`ip access-list extended` *name*	External interface: Specifies the inbound access list. or Internal interface: Specifies the outbound access list. (Using this command also causes you to enter the access-list configuration mode.)
2	`evaluate` *name*	Adds an entry that "points" to the reflexive access list. Adds an entry for each reflexive access list *name* previously defined.

Again, the order of entries is important. Normally, when a packet is evaluated against entries in an access list, the entries are evaluated in sequential order, and when a match occurs, no more entries are evaluated. With a reflexive access list nested in an extended access list, the extended access list entries are evaluated sequentially up to the nested entry, then the reflexive access list entries are evaluated sequentially, and then the remaining entries in the extended access list are evaluated sequentially. As usual, after a packet matches *any* of these entries, no more entries will be evaluated.

If the extended named IP access list you just specified has never been applied to the interface, you must also apply the extended named IP access list to the interface.

To apply the extended named IP access list to the interface, use the following command in interface configuration mode:

Command	Purpose

`ip access-group` *name* `in`
or
`ip access-group` *name* `out`
	External interface: Applies the extended access list to the interface's inbound traffic.
	Internal interface: Applies the extended access list to the interface's outbound traffic.

Setting a Global Timeout Value

Reflexive access list entries expire after no packets in the session have been detected for a certain length of time (the "timeout" period). You can specify the timeout for a particular reflexive access list when you define the reflexive access list. But if you do not specify the timeout for a given reflexive access list, the list will use the global timeout value instead.

The global timeout value is 300 seconds by default. But, you can change the global timeout to a different value at any time.

To change the global timeout value, use the following command in global configuration mode:

Command	Purpose
ip reflexive-list timeout *seconds*	Changes the global timeout value for temporary reflexive access list entries.

Reflexive Access List Configuration Examples

The following sections provide reflexive access list configuration examples:

- External Interface Configuration Example
- Internal Interface Configuration Example

External Interface Configuration Example

This example shows reflexive access lists configured for an external interface, for a topology similar to the one in Figure 10 (shown earlier in this chapter).

This configuration example permits both inbound and outbound TCP traffic at interface Serial 1, but only if the first packet (in a given session) originated from inside your network. The interface Serial 1 connects to the Internet.

Define the interface where the session-filtering configuration is to be applied:

```
interface serial 1
 description Access to the Internet via this interface
```

Apply access lists to the interface, for inbound traffic and for outbound traffic:

```
ip access-group inboundfilters in
ip access-group outboundfilters out
```

Define the outbound access list. This is the access list that evaluates all outbound traffic on interface Serial 1.

```
ip access-list extended outboundfilters
```

Define the reflexive access list *tcptraffic*. This entry permits *all* outbound TCP traffic and creates a new access list named *tcptraffic*. Also, when an outbound TCP packet is the first in a new session, a corresponding temporary entry will be automatically created in the reflexive access list *tcptraffic*.

```
permit tcp any any reflect tcptraffic
```

Define the inbound access list. This is the access list that evaluates all inbound traffic on interface Serial 1.

```
ip access-list extended inboundfilters
```

Define the inbound access list entries. This example shows BGP and Enhanced IGRP running on the interface. Also, no ICMP traffic is permitted. The last entry points to the reflexive access list. If a packet does not match the first three entries, the packet will be evaluated against all the entries in the reflexive access list *tcptraffic*.

```
permit bgp any any
permit eigrp any any
deny icmp any any
evaluate tcptraffic
```

Define the global idle timeout value for all reflexive access lists. In this example, when the reflexive access list *tcptraffic* was defined, no timeout was specified, so *tcptraffic* uses the global timeout. Therefore, if for 120 seconds there is no TCP traffic that is part of an established session, the corresponding reflexive access list entry will be removed.

```
ip reflexive-list timeout 120
```

The example configuration looks as follows:

```
interface Serial 1
 description Access to the Internet via this interface
 ip access-group inboundfilters in
 ip access-group outboundfilters out
!
ip reflexive-list timeout 120
!
ip access-list extended outboundfilters
 permit tcp any any reflect tcptraffic
!
ip access-list extended inboundfilters
 permit bgp any any
 permit eigrp any any
 deny icmp any any
 evaluate tcptraffic
```

With this configuration, before any TCP sessions have been initiated the **show access-list** EXEC command displays the following:

```
Extended IP access list inboundfilters
 permit bgp any any
 permit eigrp any any
 deny icmp any any
 evaluate tcptraffic
Extended IP access list outboundfilters
 permit tcp any any reflect tcptraffic
```

Notice that the reflexive access list does not appear in this output. This is because before any TCP sessions have been initiated, no traffic has triggered the reflexive access list, and the list is empty (has no entries). When empty, reflexive access lists do not show up in **show access-list** output.

After a Telnet connection is initiated from within your network to a destination outside of your network, the **show access-list** EXEC command displays the following:

```
Extended IP access list inboundfilters
 permit bgp any any (2 matches)
 permit eigrp any any
 deny icmp any any
 evaluate tcptraffic
Extended IP access list outboundfilters
 permit tcp any any reflect tcptraffic
Reflexive IP access list tcptraffic
 permit tcp host 172.19.99.67 eq telnet host 192.168.60.185 eq 11005 (5 matches) (time
left 115 seconds)
```

Notice that the reflexive access list *tcptraffic* now appears and displays the temporary entry generated when the Telnet session initiated with an outbound packet.

Internal Interface Configuration Example

This is an example configuration for reflexive access lists configured for an internal interface. This example has a topology similar to the one in Figure 11 (shown earlier in this chapter).

This example is similar to the previous example; the only difference between this example and the previous example is that the entries for the outbound and inbound access lists are swapped. Please refer to the previous example for more details and descriptions.

```
interface Ethernet 0
 description Access from the I-net to our Internal Network via this interface
 ip access-group inboundfilters in
 ip access-group outboundfilters out
!
ip reflexive-list timeout 120
!
ip access-list extended outboundfilters
 permit bgp any any
 permit eigrp any any
 deny icmp any any
 evaluate tcptraffic
!
ip access-list extended inboundfilters
 permit tcp any any reflect tcptraffic
```

Configuring TCP Intercept (Preventing Denial-of-Service Attacks)

This chapter describes how to configure your router to protect TCP servers from TCP SYN-flooding attacks, a type of denial-of-service attack. This is accomplished by configuring the Cisco IOS feature known as "TCP Intercept."

For a complete description of TCP Intercept commands, refer to the "TCP Intercept Commands" chapter of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

In This Chapter

This chapter has the following sections:

- About TCP Intercept
- TCP Intercept Configuration Task List
- TCP Intercept Configuration Example

About TCP Intercept

The TCP intercept feature implements software to protect TCP servers from TCP SYN-flooding attacks, which are a type of denial-of-service attack.

A SYN-flooding attack occurs when a hacker floods a server with a barrage of requests for connection. Because these messages have unreachable return addresses, the connections cannot be established. The resulting volume of unresolved open connections eventually overwhelms the server and can cause it to deny service to valid requests, thereby preventing legitimate users from connecting to a web site, accessing e-mail, using FTP service, and so on.

The TCP intercept feature helps prevent SYN-flooding attacks by intercepting and validating TCP connection requests. In intercept mode, the TCP intercept software intercepts TCP synchronization (SYN) packets from clients to servers that match an extended access list. The software establishes a connection with the client on behalf of the destination server, and if successful, establishes the connection with the server on behalf of the client and knits the two half-connections together transparently. Thus, connection attempts from unreachable hosts will never reach the server. The software continues to intercept and forward packets throughout the duration of the connection.

In the case of illegitimate requests, the software's aggressive timeouts on half-open connections and its thresholds on TCP connection requests protect destination servers while still allowing valid requests.

When establishing your security policy using TCP intercept, you can choose to intercept all requests or only those coming from specific networks or destined for specific servers. You can also configure the connection rate and threshold of outstanding connections.

You can choose to operate TCP intercept in watch mode, as opposed to intercept mode. In watch mode, the software passively watches the connection requests flowing through the router. If a connection fails to get established in a configurable interval, the software intervenes and terminates the connection attempt.

TCP options that are negotiated on handshake (such as RFC 1323 on window scaling) will not be negotiated because the TCP intercept software does not know what the server can do or will negotiate.

TCP Intercept Configuration Task List

To configure TCP intercept, perform the tasks in the following sections. The first task is required; the rest are optional.

- Enabling TCP Intercept (Required)
- Setting the TCP Intercept Mode (Optional)
- Setting the TCP Intercept Drop Mode (Optional)
- Changing the TCP Intercept Timers (Optional)
- Changing the TCP Intercept Aggressive Thresholds (Optional)
- Monitoring and Maintaining TCP Intercept (Optional)

For TCP intercept configuration examples using the commands in this chapter, refer to the "TCP Intercept Configuration Example" section at the end of this chapter.

Enabling TCP Intercept

To enable TCP intercept, use the following commands in global configuration mode:

Step	Command	Purpose
1	**access-list** *access-list-number* {**deny** | **permit**} **tcp any** *destination destination-wildcard*	Defines an IP extended access list.
2	**ip tcp intercept list** *access-list-number*	Enables TCP intercept.

You can define an access list to intercept all requests or only those coming from specific networks or destined for specific servers. Typically the access list will define the source as **any** and define specific destination networks or servers. That is, you do not attempt to filter on the source addresses because you do not necessarily know who to intercept packets from. You identify the destination in order to protect destination servers.

If no access list match is found, the router allows the request to pass with no further action.

Setting the TCP Intercept Mode

The TCP intercept can operate in either active intercept mode or passive watch mode. The default is intercept mode.

In intercept mode, the software actively intercepts each incoming connection request (SYN) and responds on behalf of the server with an SYN-ACK, then waits for an ACK from the client. When that ACK is received, the original SYN is sent to the server and the software performs a three-way handshake with the server. When this is complete, the two half-connections are joined.

In watch mode, connection requests are allowed to pass through the router to the server but are watched until they become established. If they fail to become established within 30 seconds (configurable with the **ip tcp intercept watch-timeout** command), the software sends a Reset to the server to clear up its state.

To set the TCP intercept mode, use the following command in global configuration mode:

Command	Purpose
`ip tcp intercept mode {intercept	watch}`	Sets the TCP intercept mode.

Setting the TCP Intercept Drop Mode

When under attack, the TCP intercept feature becomes more aggressive in its protective behavior. If the number of incomplete connections exceeds 1100 or the number of connections arriving in the last one minute exceeds 1100, each new arriving connection causes the oldest partial connection to be deleted. Also, the initial retransmission timeout is reduced by half to 0.5 seconds (so the total time trying to establish a connection is cut in half).

By default, the software drops the oldest partial connection. Alternatively, you can configure the software to drop a random connection. To set the drop mode, use the following command in global configuration mode:

Command	Purpose
`ip tcp intercept drop-mode {oldest	random}`	Sets the drop mode.

Changing the TCP Intercept Timers

By default, the software waits for 30 seconds for a watched connection to reach established state before sending a Reset to the server. To change this value, use the following command in global configuration mode:

Command	Purpose
`ip tcp intercept watch-timeout` *seconds*	Changes the time allowed to reach established state.

By default, the software waits for 5 seconds from receipt of a reset or FIN-exchange before it ceases to manage the connection. To change this value, use the following command in global configuration mode:

Command	Purpose
`ip tcp intercept finrst-timeout` *seconds*	Changes the time between receipt of a reset or FIN-exchange and dropping the connection.

By default, the software still manages a connection for 24 hours after no activity. To change this value, use the following command in global configuration mode:

Command	Purpose
`ip tcp intercept connection-timeout` *seconds*	Changes the time the software will manage a connection after no activity.

Changing the TCP Intercept Aggressive Thresholds

Two factors determine when aggressive behavior begins and ends: total incomplete connections and connection requests during the last one-minute sample period. Both thresholds have default values that can be redefined.

When a threshold is exceeded, the TCP intercept assumes the server is under attack and goes into aggressive mode. When in aggressive mode, the following occurs:

- Each new arriving connection causes the oldest partial connection to be deleted. (You can change to a random drop mode.)
- The initial retransmission timeout is reduced by half to 0.5 seconds, and so the total time trying to establish the connection is cut in half. (When not in aggressive mode, the code does exponential back-off on its retransmissions of SYN segments. The initial retransmission timeout is 1 second. The subsequent timeouts are 2 seconds, 4 seconds, 8 seconds, and 16 seconds. The code retransmits 4 times before giving up, so it gives up after 31 seconds of no acknowledgment.)
- If in watch mode, the watch timeout is reduced by half. (If the default is in place, the watch timeout becomes 15 seconds.)

The drop strategy can be changed from the oldest connection to a random connection with the **ip tcp intercept drop-mode** command.

Note The two factors that determine aggressive behavior are related and work together. When *either* of the **high** values is exceeded, aggressive behavior begins. When *both* quantities fall below the **low** value, aggressive behavior ends.

You can change the threshold for triggering aggressive mode based on the total number of incomplete connections. The default values for **low** and **high** are 900 and 1100 incomplete connections, respectively. To change these values, use the following commands in global configuration mode:

Step	Command	Purpose
1	`ip tcp intercept max-incomplete low` *number*	Sets the threshold for stopping aggressive mode.
2	`ip tcp intercept max-incomplete high` *number*	Sets the threshold for triggering aggressive mode.

You can also change the threshold for triggering aggressive mode based on the number of connection requests received in the last 1-minute sample period. The default values for **low** and **high** are 900 and 1100 connection requests, respectively. To change these values, use the following commands in global configuration mode:

Step	Command	Purpose
1	`ip tcp intercept one-minute low` *number*	Sets the threshold for stopping aggressive mode.
2	`ip tcp intercept one-minute high` *number*	Sets the threshold for triggering aggressive mode.

Monitoring and Maintaining TCP Intercept

To display TCP intercept information, use either of the following commands in EXEC mode:

Command	Purpose
`show tcp intercept connections`	Displays incomplete connections and established connections.
`show tcp intercept statistics`	Displays TCP intercept statistics.

TCP Intercept Configuration Example

The following configuration defines extended IP access list 101, causing the software to intercept packets for all TCP servers on the 192.168.1.0/24 subnet:

```
ip tcp intercept list 101
!
access-list 101 permit tcp any 192.168.1.0 0.0.0.255
```

Configuring Context-Based Access Control

This chapter describes how to configure Context-based Access Control (CBAC). CBAC provides advanced traffic filtering functionality and can be used as an integral part of your network's firewall. For more information regarding firewalls, refer to the chapter "Cisco Secure Integrated Software Firewall Overview."

For a complete description of the CBAC commands used in this chapter, refer to the "Context-Based Access Control Commands" chapter in the *Cisco IOS Security Command Reference.* To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

In This Chapter

This chapter has the following sections:

- CBAC Overview
- CBAC Configuration Task List
- Monitoring and Maintaining CBAC
- CBAC Configuration Examples

CBAC Overview

This section describes CBAC features and functions:

- What CBAC Does
- What CBAC Does Not Do
- How CBAC Works
- When and Where to Configure CBAC
- The CBAC Process
- Supported Protocols
- Restrictions
- Memory and Performance Impact

What CBAC Does

CBAC works to provide network protection on multiple levels using the following functions:

- Traffic Filtering
- Traffic Inspection
- Alerts and Audit Trails
- Intrusion Detection

Traffic Filtering

CBAC intelligently filters TCP and UDP packets based on application-layer protocol session information. You can configure CBAC to permit specified TCP and UDP traffic through a firewall only when the connection is initiated from within the network you want to protect. CBAC can inspect traffic for sessions that originate from either side of the firewall, and CBAC can be used for intranet, extranet, and Internet perimeters of your network.

Without CBAC, traffic filtering is limited to access list implementations that examine packets at the network layer, or at most, the transport layer. However, CBAC examines not only network layer and transport layer information but also examines the application-layer protocol information (such as FTP connection information) to learn about the state of the session. This allows support of protocols that involve multiple channels created as a result of negotiations in the control channel. Most of the multimedia protocols as well as some other protocols (such as FTP, RPC, and SQL*Net) involve multiple channels.

Using CBAC, Java blocking can be configured to filter HTTP traffic based on the server address or to completely deny access to Java applets that are not embedded in an archived or compressed file. With Java, you must protect against the risk of users inadvertently downloading destructive applets into your network. To protect against this risk, you could require all users to disable Java in their browser. If this is not an acceptable solution, you can create a CBAC inspection rule to filter Java applets at the firewall, which allows users to download only applets residing within the firewall and trusted applets from outside the firewall. For extensive content filtering of Java, Active-X, or virus scanning, you might want to consider purchasing a dedicated content filtering product.

Traffic Inspection

CBAC inspects traffic that travels through the firewall to discover and manage state information for TCP and UDP sessions. This state information is used to create temporary openings in the firewall's access lists to allow return traffic and additional data connections for permissible sessions.

Inspecting packets at the application layer, and maintaining TCP and UDP session information, provides CBAC with the ability to detect and prevent certain types of network attacks such as SYN-flooding. A SYN-flood attack occurs when a network attacker floods a server with a barrage of requests for connection and does not complete the connection. The resulting volume of half-open connections can overwhelm the server, causing it to deny service to valid requests. Network attacks that deny access to a network device are called denial-of-service (DoS) attacks.

CBAC helps to protect against DoS attacks in other ways. CBAC inspects packet sequence numbers in TCP connections to see if they are within expected ranges—CBAC drops any suspicious packets. You can also configure CBAC to drop half-open connections, which require firewall processing and memory resources to maintain. Additionally, CBAC can detect unusually high rates of new connections and issue alert messages.

CBAC can help by protecting against certain DoS attacks involving fragmented IP packets. Even though the firewall prevents an attacker from making actual connections to a given host, the attacker can disrupt services provided by that host. This is done by sending many non-initial IP fragments or by sending complete fragmented packets through a router with an ACL that filters the first fragment of a fragmented packet. These fragments can tie up resources on the target host as it tries to reassemble the incomplete packets.

Alerts and Audit Trails

CBAC also generates real-time alerts and audit trails. Enhanced audit trail features use SYSLOG to track all network transactions; recording time stamps, source host, destination host, ports used, and the total number of transmitted bytes, for advanced, session-based reporting. Real-time alerts send SYSLOG error messages to central management consoles upon detecting suspicious activity. Using CBAC inspection rules, you can configure alerts and audit trail information on a per-application protocol basis. For example, if you want to generate audit trail information for HTTP traffic, you can specify that in the CBAC rule covering HTTP inspection.

Intrusion Detection

CBAC provides a limited amount of intrusion detection to protect against specific SMTP attacks. With intrusion detection, SYSLOG messages are reviewed and monitored for specific "attack signatures." Certain types of network attacks have specific characteristics, or signatures. When CBAC detects an attacks, it resets the offending connections and sends SYSLOG information to the SYSLOG server. Refer to "Interpreting Syslog and Console Messages Generated by CBAC" for a list of supported signatures.

In addition to the limited intrusion detection offered by CBAC, the Cisco Secure Integrated Software feature set offers intrusion detection technology for mid-range and high-end router platforms using the Cisco Secure Integrated Software Intrusion Detection System (IDS). It is ideal for any network perimeter, and especially for locations in which a router is being deployed and additional security between network segments is required. It also can protect intranet and extranet connections where additional security is mandated, and branch-office sites connecting to the corporate office or Internet.

The Cisco Secure Integrated Software IDS identifies 59 of the most common attacks using signatures to detect patterns of misuse in network traffic. The intrusion-detection signatures available in the new release of the Cisco Secure Integrated Software feature set were chosen from a broad cross-section of intrusion-detection signatures. The signatures represent severe breaches of security and the most common network attacks and information-gathering scans.

For more information about Cisco Secure Integrated Software IDS, refer to the chapter "Configuring Cisco Secure Integrated Software Intrusion Detection."

What CBAC Does Not Do

CBAC does not provide intelligent filtering for all protocols; it only works for the protocols that you specify. If you do not specify a certain protocol for CBAC, the existing access lists will determine how that protocol is filtered. No temporary openings will be created for protocols not specified for CBAC inspection.

CBAC does not protect against attacks originating from within the protected network unless that traffic travels through a router that has the Cisco Secure Integrated Software feature set deployed on it. CBAC only detects and protects against attacks that travel through the firewall. This is a scenario in which you might want to deploy CBAC on an intranet-based router.

CBAC protects against certain types of attacks, but not every type of attack. CBAC should not be considered a perfect, impenetrable defense. Determined, skilled attackers might be able to launch effective attacks. While there is no such thing as a perfect defense, CBAC detects and prevents most of the popular attacks on your network.

How CBAC Works

You should understand the material in this section before you configure CBAC. If you do not understand how CBAC works, you might inadvertently introduce security risks by configuring CBAC inappropriately. This section contains the following sections:

- How CBAC Works—Overview
- How CBAC Works—Details

How CBAC Works—Overview

CBAC creates temporary openings in access lists at firewall interfaces. These openings are created when specified traffic exits your internal network through the firewall. The openings allow returning traffic (that would normally be blocked) and additional data channels to enter your internal network back through the firewall. The traffic is allowed back through the firewall only if it is part of the same session as the original traffic that triggered CBAC when exiting through the firewall.

Throughout this chapter, the terms "inbound" and "outbound" are used to describe the direction of traffic relative to the router interface on which CBAC is applied. For example, if a CBAC rule is applied inbound on interface E0, then packets entering interface E0 from the network will be inspected. If a CBAC rule is applied outbound on interface E0, then packets leaving interface E0 to the network will be inspected. This is similar to the way ACLs work.

For example, consider a CBAC inspection rule named *hqusers*, and suppose that rule is applied inbound at interface E0:

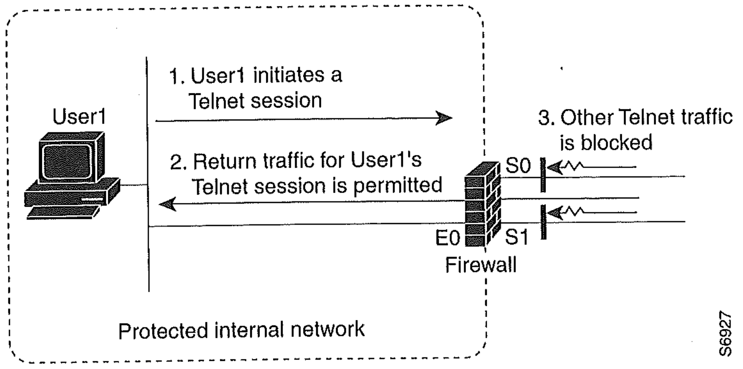
```
router (config-if)# ip inspect hqusers in
```

This command causes CBAC to inspect the packets coming into this interface from the network. If a packet is attempting to initiate a session, CBAC will then determine if this protocol is allowed, create a CBAC session, add the appropriate ACLs to allow return traffic and do any needed content inspection on any future packets for this session.

The terms "input" and "output" are used to describe the interfaces at which network traffic enters or exits the firewall router. A packet enters the firewall router via the input interface, is inspected by the firewall software and then exits the router via the output interface.

In Figure 12, the inbound access lists at S0 and S1 are configured to block Telnet traffic, and there is no outbound access list configured at E0. When the connection request for User1's Telnet session passes through the firewall, CBAC creates a temporary opening in the inbound access list at S0 to permit returning Telnet traffic for User1's Telnet session. (If the same access list is applied to both S0 and S1, the same opening would appear at both interfaces.) If necessary, CBAC would also have created a similar opening in an outbound access list at E0 to permit return traffic.

Figure 12 CBAC Opens Temporary Holes in Firewall Access Lists


User1
1. User1 initiates a Telnet session
2. Return traffic for User1's Telnet session is permitted
3. Other Telnet traffic is blocked
S0
E0
S1
Firewall
Protected internal network
S6927

How CBAC Works—Details

This section describes how CBAC inspects packets and maintains state information about sessions to provide intelligent filtering.

Packets Are Inspected

With CBAC, you specify which protocols you want to be inspected, and you specify an interface and interface direction (in or out) where inspection originates. Only specified protocols will be inspected by CBAC.

Packets entering the firewall are inspected by CBAC only if they first pass the inbound access list at the input interface and outbound access list at the output interface. If a packet is denied by the access list, the packet is simply dropped and not inspected by CBAC.

CBAC inspection tracks sequence numbers in all TCP packets, and drops those packets with sequence numbers that are not within expected ranges.

CBAC inspection recognizes application-specific commands (such as illegal SMTP commands) in the control channel, and detects and prevents certain application-level attacks.

When CBAC suspects an attack, the DoS feature can take several actions:

- Generate alert messages
- Protect system resources that could impede performance
- Block packets from suspected attackers

CBAC uses timeout and threshold values to manage session state information, helping to determine when to drop sessions that do not become fully established. Setting timeout values for network sessions helps prevent DoS attacks by freeing up system resources, dropping sessions after a specified amount of time. Setting threshold values for network sessions helps prevent DoS attacks by controlling the number of half-open sessions, which limits the amount of system resources applied to half-open sessions. When a session is dropped, CBAC sends a reset message to the devices at both end points (source and destination) of the session. When the system under DoS attack receives a reset command, it releases, or frees up, processes and resources related to that incomplete session.

CBAC provides three thresholds against DoS attacks:

- The total number of half-open TCP or UDP sessions
- The number of half-open sessions based upon time
- The number of half-open TCP-only sessions per host

If a threshold is exceeded, CBAC has two options:

- Send a reset message to the end points of the oldest half-open session, making resources available to service newly arriving SYN packets.
- In the case of half open TCP only sessions, CBAC blocks all SYN packets temporarily for the duration configured by the threshold value. When the router blocks a SYN packet, the TCP three-way handshake is never initiated, which prevents the router from using memory and processing resources needed for valid connections.

DoS detection and prevention requires that you create a CBAC inspection rule and apply that rule on an interface. The inspection rule must include the protocols that you want to monitor against DoS attacks. For example, if you have TCP inspection enabled on the inspection rule, then CBAC can track all TCP connections to watch for DoS attacks. If the inspection rule includes FTP protocol inspection but not TCP inspection, CBAC tracks only FTP connections for DoS attacks.

For detailed information about setting timeout and threshold values in CBAC to detect and prevent DoS attacks, refer in the "Configuring Global Timeouts and Thresholds" section.

A State Table Maintains Session State Information

Whenever a packet is inspected, a state table is updated to include information about the state of the session.

Return traffic will only be permitted back through the firewall if the state table contains information indicating that the packet belongs to a permissible session. CBAC controls the traffic that belongs to a valid session. When return traffic is inspected, the state table information is updated as necessary.

UDP "Sessions" Are Approximated

With UDP—a connectionless service—there are no actual sessions, so the software approximates sessions by examining the information in the packet and determining if the packet is similar to other UDP packets (for example, same source/destination addresses and port numbers) and if the packet was detected soon after another similar UDP packet. "Soon" means within the configurable UDP idle timeout period.

Access List Entries Are Dynamically Created and Deleted to Permit Return Traffic and Additional Data Connections

CBAC dynamically creates and deletes access list entries at the firewall interfaces, according to the information maintained in the state tables. These access list entries are applied to the interfaces to examine traffic flowing back into the internal network. These entries create temporary openings in the firewall to permit only traffic that is part of a permissible session.

The temporary access list entries are never saved to NVRAM.

When and Where to Configure CBAC

Configure CBAC at firewalls protecting internal networks. Such firewalls should be Cisco routers with the Cisco Secure Integrated Software feature set configured as described previously in the section "Cisco Secure Integrated Software."

Use CBAC when the firewall will be passing traffic such as:

- Standard TCP and UDP Internet applications
- Multimedia applications
- Oracle support

Use CBAC for these applications if you want the application's traffic to be permitted through the firewall only when the traffic session is initiated from a particular side of the firewall (usually from the protected internal network).

In many cases, you will configure CBAC in one direction only at a single interface, which causes traffic to be permitted back into the internal network only if the traffic is part of a permissible (valid, existing) session. This is a typical configuration for protecting your internal networks from traffic that originates on the Internet.

You can also configure CBAC in two directions at one or more interfaces. CBAC is configured in two directions when the networks on both sides of the firewall should be protected, such as with extranet or intranet configurations, and to protect against DoS attacks. For example, if the firewall is situated between two partner companies' networks, you might wish to restrict traffic in one direction for certain applications, and restrict traffic in the opposite direction for other applications.

The CBAC Process

This section describes a sample sequence of events that occurs when CBAC is configured at an external interface that connects to an external network such as the Internet.

In this example, a TCP packet exits the internal network through the firewall's external interface. The TCP packet is the first packet of a Telnet session, and TCP is configured for CBAC inspection.

1. The packet reaches the firewall's external interface.
2. The packet is evaluated against the interface's existing outbound access list, and the packet is permitted. (A denied packet would simply be dropped at this point.)
3. The packet is inspected by CBAC to determine and record information about the state of the packet's connection. This information is recorded in a new state table entry created for the new connection.

   (If the packet's application—Telnet—was not configured for CBAC inspection, the packet would simply be forwarded out the interface at this point without being inspected by CBAC. See the section "Define an Inspection Rule" for configuring CBAC inspection information.)
4. Based on the obtained state information, CBAC creates a temporary access list entry which is inserted at the beginning of the external interface's inbound extended access list. This temporary access list entry is designed to permit inbound packets that are part of the same connection as the outbound packet just inspected.
5. The outbound packet is forwarded out the interface.
6. Later, an inbound packet reaches the interface. This packet is part of the same Telnet connection previously established with the outbound packet. The inbound packet is evaluated against the inbound access list, and it is permitted because of the temporary access list entry previously created.

7 The permitted inbound packet is inspected by CBAC, and the connection's state table entry is updated as necessary. Based on the updated state information, the inbound extended access list temporary entries might be modified in order to permit only packets that are valid for the current state of the connection.

8 Any additional inbound or outbound packets that belong to the connection are inspected to update the state table entry and to modify the temporary inbound access list entries as required, and they are forwarded through the interface.

9 When the connection terminates or times out, the connection's state table entry is deleted, and the connection's temporary inbound access list entries are deleted.

In the sample process just described, the firewall access lists are configured as follows:

- An outbound IP access list (standard or extended) is applied to the external interface. This access list permits all packets that you want to allow to exit the network, including packets you want to be inspected by CBAC. In this case, Telnet packets are permitted.
- An inbound extended IP access list is applied to the external interface. This access list denies any traffic to be inspected by CBAC—including Telnet packets. When CBAC is triggered with an outbound packet, CBAC creates a temporary opening in the inbound access list to permit only traffic that is part of a valid, existing session.

  If the inbound access list had been configured to permit *all* traffic, CBAC would be creating pointless openings in the firewall for packets that would be permitted anyway.

Supported Protocols

This section provides a list of CBAC supported protocols and includes a more detailed look at support for multimedia applications, specifically RTSP and H.323.

CBAC Supported Protocols

You can configure CBAC to inspect the following types of sessions:

- All TCP sessions, regardless of the application-layer protocol (sometimes called "single-channel" or "generic" TCP inspection)
- All UDP sessions, regardless of the application-layer protocol (sometimes called "single-channel" or "generic" UDP inspection)

You can also configure CBAC to specifically inspect certain application-layer protocols. The following application-layer protocols can all be configured for CBAC:

- CU-SeeMe (only the White Pine version)
- FTP
- H.323 (such as NetMeeting, ProShare)
- HTTP (Java blocking)
- Microsoft NetShow
- UNIX R-commands (such as rlogin, rexec, and rsh)
- RealAudio
- RTSP (Real Time Streaming Protocol)
- RPC (Sun RPC, not DCE RPC)

- SMTP
- SQL*Net
- StreamWorks
- TFTP
- VDOLive

When a protocol is configured for CBAC, that protocol traffic is inspected, state information is maintained, and in general, packets are allowed back through the firewall only if they belong to a permissible session.

RTSP and H.323 Protocol Support for Multimedia Applications

CBAC supports a number of protocols for multimedia applications that require delivery of data with real-time properties such as audio and video conferencing. This support includes the following g multimedia application protocols:

- Real Time Streaming Protocol (RTSP)
- H.323 Version 2 (H.323 V2)

RTSP and H.323 V2 inspection allows clients on a protected network to receive data associated with a multimedia session from a server on an unprotected network.

RTSP Support

RTSP is the IETF standards-based protocol (RFC 2326) for control over the delivery of data with real-time properties such as audio and video streams. It is useful for large-scale broadcasts and audio or video on demand streaming, and is supported by a variety of vendor products of streaming audio and video multimedia, including Cisco IP/TV, RealNetworks RealAudio G2 Player, and Apple QuickTime 4 software.

RFC 2326 allows RTSP to run over either UDP or TCP, though CBAC currently supports only TCP-based RTSP. RTSP establishes a TCP-based control connection, or channel, between the multimedia client and server. RTSP uses this channel to control commands such as "play" and "pause" between the client and server. These control commands and responses are text-based and are similar to HTTP.

RTSP typically relies on a UDP-based data transport protocol such as standard Real-Time Transport Protocol (RTP) to open separate channels for data and for RTP Control Protocol (RTCP) messages. RTP and RTCP channels occur in pairs, with RTP being an even numbered port and RTCP being the next consecutive port. Understanding the relationship of RTP and RTCP is important for verifying session information using CBAC **show** commands.

The RTSP client uses TCP port 554 or 8554 to open a multimedia connection with a server. The data channel or data control channel (using RTCP) between the client and the server is dynamically negotiated between the client and the server using any of the high UDP ports (1024 to 65536).

CBAC uses this port information along with connection information from the client to create dynamic access control list (ACL) entries in the firewall. As TCP or UDP connections are terminated, CBAC removes these dynamic entries from the appropriate ACLs.

CBAC support for RTSP includes the following data transport modes:

- Standard Real-Time Transport Protocol (RTP)

  RTP is an IETF standard (RFC 1889) supporting delivery of real-time data such as audio and video. RTP uses the RTP Control Protocol (RTCP) for managing the delivery of the multimedia data stream. This is the normal mode of operation for Cisco IP/TV and Apple QuickTime 4 software.

- RealNetworks Real Data Transport (RDT)

  RDT is a proprietary protocol developed by RealNetworks for data transport. This mode uses RTSP for communication control and uses RDT for the data connection and retransmission of lost packets. This is the normal mode of operation for the RealServer G2 from RealNetworks.

- Interleaved (Tunnel Mode)

  In this mode, RTSP uses the control channel to tunnel RTP or RDT traffic.

- Synchronized Multimedia Integration Language (SMIL)

  SMIL is a layout language that enables the creation of multimedia presentations consisting of multiple elements of music, voice, images, text, video and graphics. This involves multiple RTSP control and data streams between the player and the servers. This mode is available only using RTSP and RDT. SMIL is a proposed specification of the World Wide Web Consortium (W3C). The RealNetworks RealServer and RealServer G2 provide support for SMIL—Cisco IP/TV and Apple QuickTime 4 do not.

H.323 Support

CBAC support for H.323 inspection includes H.323 Version 2 and H.323 Version 1. H.323 V2 provides additional options over H.323 V1, including a "fast start" option. The fast start option minimizes the delay between the time that a user initiates a connection and the time that the user gets the data (voice, video). H.323 V2 inspection is backward compatible with H.323 V1.

With H.323 V1, after a TCP connection is established between the client and server (H.225 Channel), a separate channel for media control (H.245 Channel) is opened through which multimedia channels for audit and video are further negotiated.

The H.323 V2 client opens a connection to server which is listening on port 1720. The data channel between the client and the server is dynamically negotiated using any of the high UDP ports (1024 to 65536).

CBAC uses this port information along with connection information from the client to create dynamic access control list (ACL) entries in the firewall. As TCP or UDP connections are terminated, CBAC removes these dynamic entries from the appropriate ACLs.

Restrictions

CBAC has the following restrictions:

- CBAC is available only for IP protocol traffic. Only TCP and UDP packets are inspected. (Other IP traffic, such as ICMP, cannot be inspected with CBAC and should be filtered with basic access lists instead.)
- If you reconfigure your access lists when you configure CBAC, be aware that if your access lists block TFTP traffic into an interface, you will not be able to netboot over that interface. (This is not a CBAC-specific limitation, but is part of existing access list functionality.)

- Packets with the firewall as the source or destination address are not inspected by CBAC.
- CBAC ignores ICMP Unreachable messages.
- H.323 V2 and RTSP protocol inspection supports only the following multimedia client-server applications: Cisco IP/TV, RealNetworks RealAudio G2 Player, Apple QuickTime 4.

You can use CBAC together with all the other firewall features mentioned previously in the "Cisco Secure Integrated Software Firewall Overview" chapter.

CBAC works with fast switching and process switching.

This section also discusses restrictions concerning:

- FTP Traffic and CBAC
- IPSec and CBAC Compatibility

FTP Traffic and CBAC

- With FTP, CBAC does not allow third-party connections (three-way FTP transfer).
- When CBAC inspects FTP traffic, it only allows data channels with the destination port in the range of 1024 to 65535.
- CBAC will not open a data channel if the FTP client-server authentication fails.

IPSec and CBAC Compatibility

When CBAC and IPSec are enabled on the same router, and the firewall router is an endpoint for IPSec for the particular flow, then IPSec is compatible with CBAC (that is, CBAC can do its normal inspection processing on the flow).

If the router is not an IPSec endpoint, but the packet is an IPSec packet, then CBAC will not inspect the packets because the protocol number in the IP header of the IPSec packet is not TCP or UDP. CBAC only inspects UDP and TCP packets.

Memory and Performance Impact

CBAC uses less than approximately 600 bytes of memory per connection. Because of the memory usage, you should use CBAC only when you need to. There is also a slight amount of additional processing that occurs whenever packets are inspected.

Sometimes CBAC must evaluate long access lists, which might have presented a negative impact to performance. However, this impact is avoided, because CBAC evaluates access lists using an accelerated method (CBAC hashes access lists and evaluates the hash).

CBAC Configuration Task List

To configure CBAC, perform the tasks described in the following sections. The tasks in the first seven sections are required; the task of verifying the CBAC configuration is optional.

- Picking an Interface: Internal or External (Required)
- Configuring IP Access Lists at the Interface (Required)
- Configuring Global Timeouts and Thresholds (Required)
- Defining an Inspection Rule (Required)

- Applying the Inspection Rule to an Interface (Required)
- Configuring Logging and Audit Trail (Required)
- Other Guidelines for Configuring a Firewall (Required)
- Verifying CBAC (Optional)

Following CBAC configuration, you can monitor and maintain CBAC using the information in this section.

Note If you try to configure Context-based Access Control (CBAC) but do not have a good understanding of how CBAC works, you might inadvertently introduce security risks to the firewall and to the protected network. You should be sure you understand what CBAC does before you configure CBAC.

Note As with all networking devices, protect access into the firewall by configuring passwords as described in the "Configuring Passwords and Privileges" chapter. You should also consider configuring user authentication, authorization, and accounting as described in the "Authentication, Authorization, and Accounting (AAA)" part of this guide. Additional guidelines to help you establish a good security policy can be found in the "Cisco Secure Integrated Software Overview" chapter.

For CBAC configuration examples, refer to the "CBAC Configuration Examples" section at the end of this chapter.

Picking an Interface: Internal or External

You must decide whether to configure CBAC on an internal or external interface of your firewall.

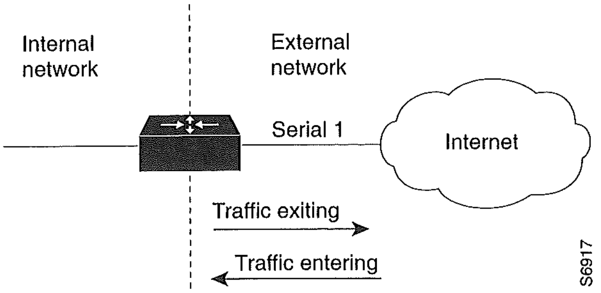
"Internal" refers to the side where sessions must originate for their traffic to be permitted through the firewall. "External" refers to the side where sessions cannot originate (sessions originating from the external side will be blocked).

If you will be configuring CBAC in two directions, you should configure CBAC in one direction first, using the appropriate "internal" and "external" interface designations. When you configure CBAC in the other direction, the interface designations will be swapped. (CBAC can be configured in two directions at one or more interfaces. Configure CBAC in two directions when the networks on both sides of the firewall require protection, such as with extranet or intranet configurations, and for protection against DoS attacks.)

The firewall is most commonly used with one of two basic network topologies. Determining which of these topologies is most like your own can help you decide whether to configure CBAC on an internal interface or on an external interface.

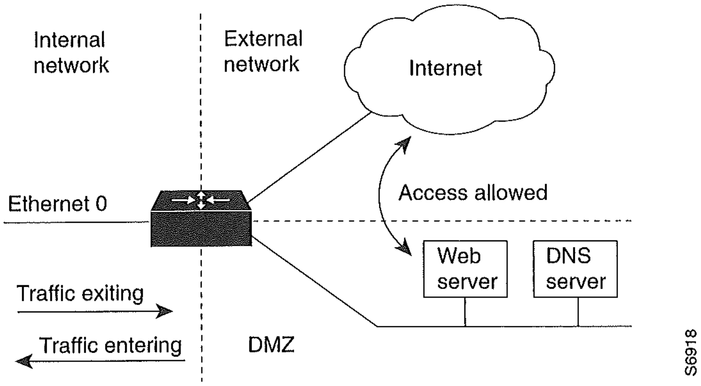
The first topology is shown in Figure 13. In this simple topology, CBAC is configured for the *external* interface Serial 1. This prevents specified protocol traffic from entering the firewall and the internal network, unless the traffic is part of a session initiated from within the internal network.

Figure 13 Simple Topology—CBAC Configured at the External Interface



The second topology is shown in Figure 14. In this topology, CBAC is configured for the *internal* interface Ethernet 0. This allows external traffic to access the services in the Demilitarized Zone (DMZ), such as DNS services, but prevents specified protocol traffic from entering your internal network—unless the traffic is part of a session initiated from within the internal network.

Figure 14 DMZ Topology—CBAC Configured at the Internal Interface



Using these two sample topologies, decide whether to configure CBAC on an internal or external interface.

To view various firewall configuration scenarios, see the "CBAC Configuration Examples" section at the end of this chapter.

Configuring IP Access Lists at the Interface

For CBAC to work properly, you need to make sure that you have IP access lists configured appropriately at the interface.

Follow these three general rules when evaluating your IP access lists at the firewall:

- Start with a basic configuration.

  If you try to configure access lists without a good understanding of how access lists work, you might inadvertently introduce security risks to the firewall and to the protected network. You should be sure you understand what access lists do before you configure your firewall. For more information about access control lists, refer to the "Access Control Lists: Overview and Guidelines" chapter.

  A basic initial configuration allows all network traffic to flow from the protected networks to the unprotected networks, while blocking network traffic from any unprotected networks.

- Permit CBAC traffic to leave the network through the firewall.

  All access lists that evaluate traffic leaving the protected network should permit traffic that will be inspected by CBAC. For example, if Telnet will be inspected by CBAC, then Telnet traffic should be permitted on all access lists that apply to traffic leaving the network.

- Use extended access lists to deny CBAC return traffic entering the network through the firewall.

  For temporary openings to be created in an access list, the access list must be an extended access list. So wherever you have access lists that will be applied to returning traffic, you must use extended access lists. The access lists should deny CBAC return traffic because CBAC will open up temporary holes in the access lists. (You want traffic to be normally blocked when it enters your network.)

Note If your firewall only has two connections, one to the internal network and one to the external network, using all inbound access lists works well because packets are stopped before they get a chance to affect the router itself.

This section contains the following sections:

- Basic Configuration
- External Interface
- Internal Interface

Basic Configuration

The first time you configure the Cisco Secure Integrated Software, it is helpful to start with a basic access list configuration that makes the operation of the firewall easy to understand without compromising security. The basic configuration allows all network traffic from the protected networks access to the unprotected networks, while blocking all network traffic (with some exceptions) from the unprotected networks to the protected networks.

Any firewall configuration depends on your site security policy. If the basic configuration does not meet your initial site security requirements, configure the firewall to meet your policy. If you are unfamiliar with that policy or need help with the configuration, contact your network administration group for assistance. For additional guidelines on configuring a firewall, refer to the "Other Guidelines for Configuring a Firewall" section in this chapter.

Use the following guidelines for configuring the initial firewall access lists:

- Do not configure an access list for traffic from the protected networks to the unprotected networks, meaning that all traffic from the protected networks can flow through the interface.

  This helps to simplify firewall management by reducing the number of access lists applied at the interfaces. Of course this assumes a high level of trust for the users on the protected networks, and it assumes there are no malicious users on the protected networks who might launch attacks from the "inside." You can fine tune network access for users on the protected networks as you gain experience with access list configuration and the operation of the firewall.

- Configure an access list that includes entries permitting certain ICMP traffic from unprotected networks.

  While an access list that denies all IP traffic not part of a connection inspected by CBAC seems most secure, it is not practical for normal operation of the router. The router expects to see ICMP traffic from other routers in the network. Additionally, ICMP traffic is not inspected by CBAC, meaning specific entries are needed in the access list to permit return traffic for ICMP commands. For example, a user on a protected network uses the **ping** command to get the status of a host on an unprotected network; without entries in the access list that permit **echo reply** messages, the user on the protected network gets no response to the **ping** command.

  Include access list entries to permit the following ICMP messages:

Message	Description
echo reply	Outgoing ping commands require echo-reply messages to come back.
time-exceeded	Outgoing traceroute commands require time-exceeded messages to come back.
packet-too-big	Path MTU discovery requires "too-big" messages to come back.
traceroute	Allow an incoming traceroute.
unreachable	Permit all "unreachable" messages to come back. If a router cannot forward or deliver a datagram, it sends an ICMP unreachable message back to the source and drops the datagram.

- Add an access list entry denying any network traffic from a source address matching an address on the protected network.

  This is known as anti-spoofing protection because it prevents traffic from an unprotected network from assuming the identity of a device on the protected network.

- Add an entry denying broadcast messages with a source address of 255.255.255.255.

  This entry helps to prevent broadcast attacks.

- By default, the last entry in an extended access list is an implicit denial of all IP traffic not specifically allowed by other entries in the access list.

  Although this is the default setting, this final deny statement is not shown by default in an access list. Optionally, you can add an entry to the access list denying IP traffic with any source or destination address with no undesired effects.

For complete information about how to configure IP access lists, refer to the "Configuring IP Services" chapter of the *Cisco IOS IP and IP Routing Configuration Guide*.

For tips on applying access lists at an external or internal interface, review the sections "External Interface" and "Internal Interface" in this chapter.

External Interface

Here are some tips for your access lists when you will be configuring CBAC on an external interface:

- If you have an outbound IP access list at the external interface, the access list can be a standard or extended access list. This outbound access list should permit traffic that you want to be inspected by CBAC. If traffic is not permitted, it will not be inspected by CBAC, but will be simply dropped.
- The inbound IP access list at the external interface must be an extended access list. This inbound access list should deny traffic that you want to be inspected by CBAC. (CBAC will create temporary openings in this inbound access list as appropriate to permit only return traffic that is part of a valid, existing session.)
- For complete information about how to configure IP access lists, refer to the "Configuring IP Services" chapter of the *Cisco IOS IP and IP Routing Configuration Guide.*

Internal Interface

Here are some tips for your access lists when you will be configuring CBAC on an internal interface:

- If you have an inbound IP access list at the internal interface or an outbound IP access list at external interface(s), these access lists can be either a standard or extended access list. These access lists should permit traffic that you want to be inspected by CBAC. If traffic is not permitted, it will not be inspected by CBAC, but will be simply dropped.
- The outbound IP access list at the internal interface and the inbound IP access list at the external interface must be extended access lists. These outbound access lists should deny traffic that you want to be inspected by CBAC. (CBAC will create temporary openings in these outbound access lists as appropriate to permit only return traffic that is part of a valid, existing session.) You do not necessarily need to configure an extended access list at both the outbound internal interface and the inbound external interface, but at least one is necessary to restrict traffic flowing through the firewall into the internal protected network.
- For complete information about how to configure IP access lists, refer to the "Configuring IP Services" chapter of the *Cisco IOS IP and IP Routing Configuration Guide.*

Configuring Global Timeouts and Thresholds

CBAC uses timeouts and thresholds to determine how long to manage state information for a session, and to determine when to drop sessions that do not become fully established. These timeouts and thresholds apply globally to all sessions.

You can use the default timeout and threshold values, or you can change to values more suitable to your security requirements. You should make any changes to the timeout and threshold values before you continue configuring CBAC.

Note If you want to enable the more aggressive TCP host-specific denial-of-service prevention that includes the blocking of connection initiation to a host, you must set the **block-time** specified in the **ip inspect tcp max-incomplete host** command (see the last row in Table 17).

All the available CBAC timeouts and thresholds are listed in Table 17, along with the corresponding command and default value. To change a global timeout or threshold listed in the "Timeout of Threshold Value to Change" column, use the global configuration command in the "Command" column:

Table 17 Timeout and Threshold Values

Timeout or Threshold Value to Change	Command	Default
The length of time the software waits for a TCP session to reach the established state before dropping the session.	**ip inspect tcp synwait-time** *seconds*	30 seconds
The length of time a TCP session will still be managed after the firewall detects a FIN-exchange.	**ip inspect tcp finwait-time** *seconds*	5 seconds
The length of time a TCP session will still be managed after no activity (the TCP idle timeout).[1]	**ip inspect tcp idle-time** *seconds*	3600 seconds (1 hour)
The length of time a UDP session will still be managed after no activity (the UDP idle timeout).[1]	**ip inspect udp idle-time** *seconds*	30 seconds
The length of time a DNS name lookup session will still be managed after no activity.	**ip inspect dns-timeout** *seconds*	5 seconds
The number of existing half-open sessions that will cause the software to start deleting half-open sessions.[2]	**ip inspect max-incomplete high** *number*	500 existing half-open sessions
The number of existing half-open sessions that will cause the software to stop deleting half-open sessions.[2]	**ip inspect max-incomplete low** *number*	400 existing half-open sessions
The rate of new sessions that will cause the software to start deleting half-open sessions.[2]	**ip inspect one-minute high** *number*	500 half-open sessions per minute
The rate of new sessions that will cause the software to stop deleting half-open sessions.[2]	**ip inspect one-minute low** *number*	400 half-open sessions per minute
The number of existing half-open TCP sessions with the same destination host address that will cause the software to start dropping half-open sessions to the same destination host address.[3]	**ip inspect tcp max-incomplete host** *number* **block-time** *minutes*	50 existing half-open TCP sessions; 0 minutes

[1] The global TCP and UDP idle timeouts can be overridden for specified application-layer protocols' sessions as described in the **ip inspect name** (global configuration) command description, found in the "Context-Based Access Control Commands" chapter.

[2] See the following section, "Half-Open Sessions," for more information.

[3] Whenever the **max-incomplete host** threshold is exceeded, the software will drop half-open sessions differently depending on whether the **block-time** timeout is zero or a positive non-zero number. If the **block-time** timeout is zero, the software will delete the oldest existing half-open session for the host for every new connection request to the host and will let the SYN packet through. If the **block-time** timeout is greater than zero, the software will delete all existing half-open sessions for the host, and then block all new connection requests to the host. The software will continue to block all new connection requests until the **block-time** expires.

To reset any threshold or timeout to the default value, use the **no** form of the command in Table 17.

Half-Open Sessions

An unusually high number of half-open sessions (either absolute or measured as the arrival rate) could indicate that a denial-of-service attack is occurring. For TCP, "half-open" means that the session has not reached the established state—the TCP three-way handshake has not yet been completed. For UDP, "half-open" means that the firewall has detected no return traffic.

CBAC measures both the total number of existing half-open sessions and the rate of session establishment attempts. Both TCP and UDP half-open sessions are counted in the total number and rate measurements. Rate measurements are made several times per minute.

When the number of existing half-open sessions rises above a threshold (the **max-incomplete high** number), the software will delete half-open sessions as required to accommodate new connection requests. The software will continue to delete half-open requests as necessary, until the number of existing half-open sessions drops below another threshold (the **max-incomplete low** number).

When the rate of new connection attempts rises above a threshold (the **one-minute high** number), the software will delete half-open sessions as required to accommodate new connection attempts. The software will continue to delete half-open sessions as necessary, until the rate of new connection attempts drops below another threshold (the **one-minute low** number). The rate thresholds are measured as the number of new session connection attempts detected in the last one-minute sample period.

Defining an Inspection Rule

After you configure global timeouts and thresholds, you must define an inspection rule. This rule specifies what IP traffic (which application-layer protocols) will be inspected by CBAC at an interface.

Normally, you define only one inspection rule. The only exception might occur if you want to enable CBAC in two directions as described earlier in the section "When and Where to Configure CBAC." For CBAC configured in both directions at a single firewall interface, you should configure two rules, one for each direction.

An inspection rule should specify each desired application-layer protocol as well as generic TCP or generic UDP if desired. The inspection rule consists of a series of statements each listing a protocol and specifying the same inspection rule name.

Inspection rules include options for controlling alert and audit trail messages and for checking IP packet fragmentation.

To define an inspection rule, follow the instructions in the following sections:

- Configuring Application-Layer Protocol Inspection
- Configuring Generic TCP and UDP Inspection

Configuring Application-Layer Protocol Inspection

This section provides instructions for configuring CBAC with the following inspection information:

- Configuring Application-Layer Protocols
- Configuring Java Blocking
- Configuring IP Packet Fragmentation Inspection

Note For CBAC inspection to work with NetMeeting 2.0 traffic (an H.323 application-layer protocol), you must also configure inspection for TCP, as described later in the "Configuring Generic TCP and UDP Inspection" section. This requirement exists because NetMeeting 2.0 uses an additional TCP channel not defined in the H.323 specification.

Configuring Application-Layer Protocols

To configure CBAC inspection for an application-layer protocol, use one or both of the following commands in global configuration mode:

Command	Purpose
Router(config)# **ip inspect name** *inspection-name* *protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*]	Configures CBAC inspection for an application-layer protocol (except for RPC and Java). Use one of the protocol keywords defined in Table 18. Repeat this command for each desired protocol. Use the same *inspection-name* to create a single inspection rule.
Router(config)# **ip inspect name** *inspection-name* **rpc program-number** *number* [**wait-time** *minutes*] [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*]	Enables CBAC inspection for the RPC application-layer protocol. You can specify multiple RPC program numbers by repeating this command for each program number. Use the same *inspection-name* to create a single inspection rule.

Refer to the description of the **ip inspect name** global configuration command in the "Context-Based Access Control Commands" chapter of the *Cisco IOS Security Command Reference* for more information about how the command works with each application-layer protocol.

To enable CBAC inspection for Java blocking, see the following section, "Configuring Java Blocking."

Table 18 identifies application protocol keywords for the **ip inspect name** command.

Table 18 Application Protocol Keywords for the ip inspect name Command

Application Protocol	*protocol* Keyword
CU-SeeMe	**cuseeme**
FTP	**ftp**
H.323	**h323**
Microsoft NetShow	**netshow**
UNIX R commands (rlogin, rexec, rsh)	**rcmd**
RealAudio	**realaudio**
SMTP	**smtp**
SQL*Net	**sqlnet**
StreamWorks	**streamworks**
TFTP	**tftp**
VDOLive	**vdolive**

Configuring Java Blocking

Java applet filtering distinguishes between trusted and untrusted applets by relying on a list of external sites that you designate as "friendly." If an applet is from a friendly site, the firewall allows the applet through. If the applet is not from a friendly site, the applet will be blocked. (Alternately, you could permit applets from all external sites except for those you specifically designate as hostile.)

To block all Java applets except for applets from friendly locations, use the following commands in global configuration mode:

Step	Command	Purpose
1	Router(config)# **ip access-list standard** *name* **permit** ... **deny** ... (Use permit and deny statements as appropriate.) or Router(config)# **access-list** *access-list-number* {**deny** | **permit**} *protocol source* [*source-wildcard*]eq www *destination* [*destination-wildcard*]	Creates a standard access list that permits traffic only from friendly sites, and denies traffic from hostile sites. Use the **any** keyword for the destination as appropriate—but be careful to not misuse the **any** keyword to inadvertently allow all applets through.
2	Router(config)# **ip inspect name** **inspection-name** **http** [**java-list** *access-list*] [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*]	Blocks all Java applets except for applets from the friendly sites defined previously in the access list. Java blocking only works with standard access lists. Use the same *inspection-name* as when you specified other protocols, to create a single inspection rule.

Caution CBAC does not detect or block encapsulated Java applets. Therefore, Java applets that are wrapped or encapsulated, such as applets in .zip or .jar format, are *not* blocked at the firewall. CBAC also does not detect or block applets loaded from FTP, gopher, HTTP on a nonstandard port, and so forth.

Configuring IP Packet Fragmentation Inspection

CBAC inspection rules can help protect hosts against certain DoS attacks involving fragmented IP packets.

Using fragmentation inspection, the firewall maintains an *interfragment state* (structure) for IP traffic. Non-initial fragments are discarded unless the corresponding initial fragment was permitted to pass through the firewall. Non-initial fragments received before the corresponding initial fragments are discarded.

Note Fragmentation inspection can have undesirable effects in certain cases, because it can result in the firewall discarding any packet whose fragments arrive out of order. There are many circumstances that can cause out-of-order delivery of legitimate fragments. Applying fragmentation inspection in situations where legitimate fragments, which are likely to arrive out of order, might have a severe performance impact.

Because routers running Cisco IOS software are used in a large variety of networks, and because the CBAC feature is often used to isolate parts of internal networks from one another, the fragmentation inspection feature is disabled by default. Fragmentation detection must be explicitly enabled for an inspection rule using the **ip inspect name** command. Unfragmented traffic is never discarded because it lacks a fragment state. Even when the system is under heavy attack with fragmented packets, legitimate fragmented traffic, if any, gets some fraction of the firewall's fragment state resources, and legitimate, unfragmented traffic can flow through the firewall unimpeded.

Configuring Generic TCP and UDP Inspection

You can configure TCP and UDP inspection to permit TCP and UDP packets to enter the internal network through the firewall, even if the application-layer protocol is not configured to be inspected. However, TCP and UDP inspection do not recognize application-specific commands, and therefore might not permit all return packets for an application, particularly if the return packets have a different port number than the previous exiting packet.

Any application-layer protocol that is inspected will take precedence over the TCP or UDP packet inspection. For example, if inspection is configured for FTP, all control channel information will be recorded in the state table, and all FTP traffic will be permitted back through the firewall if the control channel information is valid for the state of the FTP session. The fact that TCP inspection is configured is irrelevant to the FTP state information.

With TCP and UDP inspection, packets entering the network must exactly match the corresponding packet that previously exited the network. The entering packets must have the same source/destination addresses and source/destination port numbers as the exiting packet (but reversed); otherwise, the entering packets will be blocked at the interface. Also, all TCP packets with a sequence number outside of the window are dropped.

With UDP inspection configured, replies will only be permitted back in through the firewall if they are received within a configurable time after the last request was sent out. (This time is configured with the **ip inspect udp idle-time** command.)

To configure CBAC inspection for TCP or UDP packets, use one or both of the following commands in global configuration mode:

Command	Purpose
`Router(config)# ip inspect name inspection-name tcp [alert {on	off}] [audit-trail {on	off}] [timeout seconds]`	Enables CBAC inspection for TCP packets. Use the same *inspection-name* as when you specified other protocols, to create a single inspection rule.
`Router(config)# ip inspect name inspection-name udp [alert {on	off}] [audit-trail {on	off}] [timeout seconds]`	Enables CBAC inspection for UDP packets. Use the same *inspection-name* as when you specified other protocols, to create a single inspection rule.

Applying the Inspection Rule to an Interface

After you define an inspection rule, you apply this rule to an interface.

Normally, you apply only one inspection rule to one interface. The only exception might occur if you want to enable CBAC in two directions as described earlier in the section "When and Where to Configure CBAC." For CBAC configured in both directions at a single firewall interface, you should apply two rules, one for each direction.

If you are configuring CBAC on an external interface, apply the rule to outbound traffic.

If you are configuring CBAC on an internal interface, apply the rule to inbound traffic.

To apply an inspection rule to an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip inspect inspection-name {in | out}`	Applies an inspection rule to an interface.

Configuring Logging and Audit Trail

Turn on logging and audit trail to provide a record of network access through the firewall, including illegitimate access attempts, and inbound and outbound services. To configure logging and audit trail functions, enter the following commands in global configuration mode:

Step	Command	Purpose
1	`Router(config)# service timestamps log datetime`	Adds the date and time to syslog and audit trail messages.
2	`Router(config)# logging host`	Specifies the host name or IP address of the host where you want to send syslog messages.
3	`Router(config)# logging facility facility-type`	Configures the syslog facility in which error messages are sent.
4	`Router(config)# logging trap level`	(Optional) Uses this command to limit messages logged to the syslog servers based on severity. The default is level 7 (informational).
5	`Router(config)# ip inspect audit-trail`	Turns on CBAC audit trail messages.

For information on how to interpret the syslog and audit trail messages, refer to the "Interpreting Syslog and Console Messages Generated by CBAC" section.

To configure audit trail functions on a per-application basis, refer to the "Defining an Inspection Rule" section for more information.

For complete information about how to configure logging, refer to the "Troubleshooting the Router" chapter of the *Cisco IOS Configuration Fundamentals Configuration Guide.*

Other Guidelines for Configuring a Firewall

As with all networking devices, you should always protect access into the firewall by configuring passwords as described in the "Configuring Passwords and Privileges" chapter. You should also consider configuring user authentication, authorization, and accounting as described in the "Authentication, Authorization, and Accounting (AAA)" part of this guide.

You should also consider the following recommendations:

- When setting passwords for privileged access to the firewall, use the **enable secret** command rather than the **enable password** command, which does not have as strong an encryption algorithm.
- Put a password on the console port. In authentication, authorization, and accounting (AAA) environments, use the same authentication for the console as for elsewhere. In a non-AAA environment, at a minimum configure the **login** and **password** *password* commands.
- Think about access control *before* you connect a console port to the network in any way, including attaching a modem to the port. Be aware that a *break* on the console port might give total control of the firewall, even with access control configured.

- Apply access lists and password protection to all virtual terminal ports. Use access lists to limit who can Telnet into your router.
- Do not enable any local service (such as SNMP or NTP) that you do not use. Cisco Discovery Protocol (CDP) and Network Time Protocol (NTP) are on by default, and you should turn these off if you do not need them.

  To turn off CDP, enter the **no cdp run** global configuration command. To turn off NTP, enter the **ntp disable** interface configuration command on each interface not using NTP.

  If you must run NTP, configure NTP only on required interfaces, and configure NTP to listen only to certain peers.

  Any enabled service could present a potential security risk. A determined, hostile party might be able to find creative ways to misuse the enabled services to access the firewall or the network.

  For local services that are enabled, protect against misuse. Protect by configuring the services to communicate only with specific peers, and protect by configuring access lists to deny packets for the services at specific interfaces.
- Protect against spoofing: protect the networks on both sides of the firewall from being spoofed from the other side. You could protect against spoofing by configuring input access lists at all interfaces to pass only traffic from expected source addresses, and to deny all other traffic.

  You should also disable source routing. For IP, enter the **no ip source-route** global configuration command. Disabling source routing at *all* routers can also help prevent spoofing.

  You should also disable minor services. For IP, enter the **no service tcp-small-servers** and **no service udp-small-servers** global configuration commands. In Cisco IOS Release 12.0 and later, these services are disabled by default.
- Prevent the firewall from being used as a relay by configuring access lists on any asynchronous Telnet ports.
- Normally, you should disable directed broadcasts for all applicable protocols on your firewall and on all your other routers. For IP, use the **no ip directed-broadcast** command. Rarely, some IP networks do require directed broadcasts; if this is the case, do not disable directed broadcasts.

  Directed broadcasts can be misused to multiply the power of denial-of-service attacks, because every denial-of-service packet sent is broadcast to every host on a subnet. Furthermore, some hosts have other intrinsic security risks present when handling broadcasts.
- Configure the **no proxy-arp** command to prevent internal addresses from being revealed. (This is important to do if you do not already have NAT configured to prevent internal addresses from being revealed).
- Keep the firewall in a secured (locked) room.

Verifying CBAC

You can view and verify CBAC configuration, status, statistics, and session information by using one or more of the following commands in EXEC mode:

Command	Purpose
`Router# show ip access-lists`	Displays the contents of all current IP access lists.
`Router# show ip inspect name inspection-name`	Shows a particular configured inspection rule.
`Router# show ip inspect config`	Shows the complete CBAC inspection configuration.
`Router# show ip inspect interfaces`	Shows interface configuration with regards to applied inspection rules and access lists.

Command	Purpose
`Router# show ip inspect session [detail]`	Shows existing sessions that are currently being tracked and inspected by CBAC.
`Router# show ip inspect all`	Shows all CBAC configuration and all existing sessions that are currently being tracked and inspected by CBAC.

In most cases, you can tell whether CBAC is inspecting network traffic properly because network applications are working as expected. In some cases, however, you might want to verify CBAC operation. For example, to verify RTSP or H.323 inspection, initiate an RTSP- or H.323-based application through the firewall. Use the **show ip inspect session** and **show ip access lists** commands to verify CBAC operation. These commands display the dynamic ACL entries and the established connections for a multimedia session.

In the case of RTSP inspection, session output can vary based on the multimedia protocol and the transport mode. This section uses examples of RTSP and H.323 V2 sessions to illustrate verification procedures and to illustrate how session information, and the interpretation of that session information, varies based on the protocol being inspected. This section provides the following sample session output:

- RTSP with RDT
- RTSP with TCP Only (Interleaved Mode)
- RTSP with SMIL
- RTSP with RTP (IP/TV)
- H.323 V2

RTSP with RDT

The following example illustrates the result of the **show ip inspect session** command. It shows that a control channel (rtsp) and data channel (rtsp-data) are open between hosts 192.168.155.2 and 192.168.35.1:

```
router#sh ip inspect sessions
Established Sessions
 Session 616B4F1C (192.168.155.2:7548)=>(192.168.35.1:6970) rtsp-data SIS_OPEN
 Session 611E2904 (192.168.35.1:1221)=>(192.168.155.2:554) rtsp SIS_OPEN
```

The following example illustrates the result of the **show ip access-list** command. It shows that two dynamic entries (permit statements) were added to ACL 100 for the multimedia session. The TCP entry creates a dynamic opening through the firewall between port 554 (RTSP protocol port) on the client and port 1221 on the server. The UDP entry creates a dynamic opening between data port 7548 on the client and data port 6970 on the server.

```
router#sh ip access-list
Extended IP access list 100
    permit udp host 192.168.155.2 eq 7548 host 192.168.35.1 eq 6970 (31 matches)
    permit tcp host 192.168.155.2 eq 554 host 192.168.35.1 eq 1221 (27 matches)
```

After closing the multimedia session, review the session output using the **show** commands to verify the firewall software has removed the dynamic entries from the configuration.

RTSP with TCP Only (Interleaved Mode)

The following example illustrates the result of the **show ip inspect session** command. It shows that only a single control channel (rtsp) is open between hosts 192.168.155.2 and 192.168.35.1. In this mode, data is tunneled through the firewall using the TCP connection to interleave RDT or RTP data.

```
router#sh ip inspect sessions
Established Sessions
 Session 611E2904 (192.168.35.1:1228)=>(192.168.155.2:554) rtsp SIS_OPEN
```

The following example illustrates the result of the **show ip access-list** command. It shows that a single dynamic entry (permit statement) was added to ACL 100 for the multimedia session. The TCP entry creates a dynamic opening through the firewall between port 554 (RTSP protocol port) on the client and port 1228 on the server.

```
router#sh ip access-lists
Extended IP access list 100
    permit tcp host 192.168.155.2 eq 554 host 192.168.35.1 eq 1228 (391 matches)
```

After closing the multimedia session, review the session output using the **show** commands to verify the firewall software has removed the dynamic entries from the configuration.

RTSP with SMIL

The following example illustrates the result of the **show ip inspect session** command for RTSP using Synchronized Multimedia Integration Language (SMIL). It shows that a single control channel (rtsp) and multiple data channels (rtsp-data) are open between hosts 192.168.155.2 and 192.168.35.1. The data channels appear as half open sessions because the UDP data flows in one direction only, which is from the server to the client.

```
router#sh ip inspect session
Established Sessions
   Session 616CA914 (192.168.155.2:30616)=>(192.168.35.1:6974) rtsp-data SIS_OPEN
   Session 616B4E78 (192.168.35.1:1230)=>(192.168.155.2:554) rtsp SIS_OPEN
   Session 614AB61C (192.168.155.2:29704)=>(192.168.35.1:6976) rtsp-data SIS_OPEN
   Session 616CAA88 (192.168.155.2:26764)=>(192.168.35.1:6972) rtsp-data SIS_OPEN
Half-open Sessions
   Session 614AAEF0 (192.168.155.2:15520)=>(192.168.35.1:6970) rtsp-data SIS_OPENING
```

The following example illustrates the result of the **show ip access-lists** command. It shows that multiple dynamic entries (permit statements) were added to ACL 100 for the multimedia session. The TCP entry creates a dynamic opening through the firewall between port 554 (RTSP protocol port) on the client and port 1230 on the server. The UDP entries create dynamic openings between negotiated data ports on the client (192.168.155.2) and the server (192.168.35.1).

```
router#sh ip access-lists
Extended IP access list 100
 permit udp host 192.168.155.2 eq 29704 host 192.168.35.1 eq 6976 (182 matches)
 permit udp host 192.168.155.2 eq 30616 host 192.168.35.1 eq 6974 (268 matches)
 permit udp host 192.168.155.2 eq 26764 host 192.168.35.1 eq 6972 (4 matches)
 permit udp host 192.168.155.2 eq 15520 host 192.168.35.1 eq 6970 (12 matches)
 permit tcp host 192.168.155.2 eq 554 host 192.168.35.1 eq 1230 (41 matches)
```

After closing the multimedia session, review the session output using the **show** commands to verify the firewall software has removed the dynamic entries from the configuration.

RTSP with RTP (IP/TV)

The following example illustrates the result of the **show ip inspect session** command for RTSP with the Cisco IP/TV application. The output shows that a single control channel (rtsp) and multiple data channels (rtsp-data) are open between hosts 192.168.2.15 and 192.168.102.23. The data channels appear as half-open sessions because the UDP data flows in one direction only, which is from the server to the client.

```
router# show ip inspect sessions
Established Sessions
 Session 611493C0 (192.168.2.15:2571)=>(192.168.102.23:8554) rtsp SIS_OPEN
Half-open Sessions
 Session 6114A22C (192.168.102.23:2428)=>(192.168.2.15:20112) rtsp-data SIS_OPENING
 Session 61149F44 (192.168.102.23:2428)=>(192.168.2.15:20113) rtsp-data SIS_OPENING
 Session 6114A0B8 (192.168.102.23:2429)=>(192.168.2.15:20115) rtsp-data SIS_OPENING
 Session 6114A3A0 (192.168.102.23:2429)=>(192.168.2.15:20114) rtsp-data SIS_OPENING
```

The following example illustrates the result of the **show ip access-lists** command. It shows that multiple dynamic entries (permit statements) were added to ACL 100 for the multimedia session. The TCP entry creates a dynamic opening through the firewall between port 554 (RTSP protocol port) on the client and port 1230 on the server. The UDP entries create dynamic openings between negotiated data ports on the client (192.168.2.15) and the server (192.168.102.23).

```
router# show ip access-lists
Extended IP access list 100
 permit udp host 192.168.102.23 eq 2428 host 192.168.2.15 eq 20113 (11 matches)
 permit udp host 192.168.102.23 eq 2428 host 192.168.2.15 eq 20112 (256 matches)
 permit udp host 192.168.102.23 eq 2429 host 192.168.2.15 eq 20115 (11 matches)
 permit udp host 192.168.102.23 eq 2429 host 192.168.2.15 eq 20114 (4598 matches)
 permit tcp host 192.168.102.23 eq 8554 host 192.168.2.15 eq 2571 (22 matches)
```

After closing the multimedia session, review the session output using the **show** commands to verify that the firewall software has removed the dynamic entries from the configuration.

H.323 V2

The following example illustrates the result of the **show ip inspect session** command for H.323 V2. It shows a single H.323 control channel, an RTP Control Protocol channel for both audio and video data, and an RTP data channel between hosts 192.168.155.2 and 192.168.35.1.

```
Session 615E2688 (192.168.35.1:49609)=>(192.168.155.1:49609) H323-RTCP-audio SIS_OPEN
Session 615E2688 (192.168.35.1:49508)=>(192.168.155.1:49508) H323-RTP-audio SIS_OPEN
Session 615E2688 (192.168.35.1:49410)=>(192.168.155.1:49410) H323-RTP-video SIS_OPEN
Session 615E2688 (192.168.35.1:49611)=>(192.168.155.1:49611) H323-RTCP-video SIS_OPEN
Session 615E1640 (192.168.35.1:4414)=>(192.168.155.1:1720) H323 SIS_OPEN
```

The following example illustrates the result of the **show ip access-lists** command. It shows that multiple dynamic entries (permit statements) were added to ACL 100 for the multimedia session. The TCP entry creates a dynamic opening through the firewall between port 1720 (H.323 V2 protocol port) on the client and port 4414 on the server. The UDP entries create dynamic openings between negotiated data ports on the client (192.168.155.1) and the server (192.168.35.1).

```
router# show ip access-lists
Extended IP access list 100
 permit udp host 192.168.155.1 eq 49609 host 192.168.35.1 eq 49609 (11 matches)
 permit udp host 192.168.155.1 eq 49508 host 192.168.35.1 eq 49508 (256 matches)
 permit udp host 192.168.155.1 eq 49411 host 192.168.35.1 eq 49411 (11 matches)
 permit udp host 192.168.155.1 eq 49610 host 192.168.35.1 eq 49610 (4598 matches)
 permit tcp host 192.168.155.1 eq 1720 host 192.168.35.1 eq 4414 (22 matches)
```

Monitoring and Maintaining CBAC

You can watch for network attacks and investigate network problems using debug commands and system messages. This section has the following sections:

- Debugging Context-Based Access Control
- Interpreting Syslog and Console Messages Generated by CBAC
- Turning Off CBAC

Debugging Context-Based Access Control

To assist CBAC debugging, you can turn on audit trail messages that will be displayed on the console after each CBAC session closes. Audit trail information is also configurable on a per-application basis using the CBAC inspection rules.

To turn on audit trail messages, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip inspect audit-trail`	Turns on CBAC audit trail messages.

If required, you can also use the CBAC **debug** commands listed in this section. (Debugging can be turned off for each of the commands in this section by using the **no** form of the command. To disable all debugging, use the privileged EXEC commands **no debug all** or **undebug all.**)

The following **debug** commands are available:

- Generic Debug Commands
- Transport Level Debug Commands
- Application Protocol Debug Commands

For a complete description of the debug commands, refer to the *Cisco IOS Debug Command Reference*.

Generic Debug Commands

You can use the following generic **debug** commands, entered in privileged EXEC mode:

Command	Purpose
`Router# debug ip inspect function-trace`	Displays messages about software functions called by CBAC.
`Router# debug ip inspect object-creation`	Displays messages about software objects being created by CBAC. Object creation corresponds to the beginning of CBAC-inspected sessions.
`Router# debug ip inspect object-deletion`	Displays messages about software objects being deleted by CBAC. Object deletion corresponds to the closing of CBAC-inspected sessions.
`Router# debug ip inspect events`	Displays messages about CBAC software events, including information about CBAC packet processing.
`Router# debug ip inspect timers`	Displays messages about CBAC timer events such as when a CBAC idle timeout is reached.
`Router# debug ip inspect detail`	Enables the detailed option, which can be used in combination with other options to get additional information.

Transport Level Debug Commands

You can use the following transport-level **debug** commands, entered in privileged EXEC mode:

Command	Purpose
`Router# debug ip inspect tcp`	Displays messages about CBAC-inspected TCP events, including details about TCP packets.
`Router# debug ip inspect udp`	Displays messages about CBAC-inspected UDP events, including details about UDP packets.

Application Protocol Debug Commands

You can use the following application protocol **debug** command, entered in privileged EXEC mode:

Command	Purpose
Router# **debug ip inspect** *protocol*	Displays messages about CBAC-inspected protocol events, including details about the protocol's packets. Refer to Table 19 to determine the protocol keyword.

Table 19 identifies application protocol keywords for the **debug ip inspect** command.

Table 19 Application Protocol Keywords for the debug ip inspect Command

Application Protocol	*protocol* Keyword
CU-SeeMe	**cuseeme**
FTP commands and responses	**ftp-cmd**
FTP token (enables tracing of the FTP tokens parsed)	**ftp-token**
H.323	**h323**
HTTP (Java applets)	**http**
Microsoft NetShow	**netshow**
UNIX R commands (rlogin, rexec, rsh)	**rcmd**
RealAudio	**realaudio**
RPC	**rpc**
SMTP	**smtp**
SQL*Net	**sqlnet**
StreamWorks	**streamworks**
TFTP	**tftp**
VDOLive	**vdolive**

Interpreting Syslog and Console Messages Generated by CBAC

CBAC provides syslog messages, console alert messages, and audit trail messages. These messages are useful because they can alert you to network attacks and because they provide an audit trail that provides details about sessions inspected by CBAC. While they are generally referred to as error messages, not all error messages indicate problems with your system.

Audit trail and alert information is configurable on a per-application basis using the CBAC inspection rules.

The following types of messages can be generated by CBAC:

- Denial-of-Service Attack Detection Error Messages
- SMTP Attack Detection Error Messages
- Java Blocking Error Messages
- FTP Error Messages
- Audit Trail Messages

For explanations and recommended actions related to the error messages mentioned in this section, refer to the *Cisco IOS System Error Messages*.

Denial-of-Service Attack Detection Error Messages

CBAC detects and blocks denial-of-service attacks and notifies you when denial-of-service attacks occur. Error messages such as the following may indicate that denial-of-service attacks have occurred:

```
%FW-4-ALERT_ON: getting aggressive, count (550/500) current 1-min rate: 250
%FW-4-ALERT_OFF: calming down, count (0/400) current 1-min rate: 0
```

When `%FW-4-ALERT_ON` and `%FW-4-ALERT_OFF` error messages appear together, each "aggressive/calming" pair of messages indicates a separate attack. The preceding example shows one separate attack.

Error messages such as the following may indicate that a denial-of-service attack has occurred on a specific TCP host:

```
%FW-4-HOST_TCP_ALERT_ON: Max tcp half-open connections (50) exceeded for host
172.21.127.242.
%FW-4-BLOCK_HOST: Blocking new TCP connections to host 172.21.127.242 for 2 minutes
(half-open count 50 exceeded)
%FW-4-UNBLOCK_HOST: New TCP connections to host 172.21.127.242 no longer blocked
```

SMTP Attack Detection Error Messages

CBAC detects and blocks SMTP attacks (illegal SMTP commands) and notifies you when SMTP attacks occur. Error messages such as the following may indicate that an SMTP attack has occurred:

```
%FW-4-SMTP_INVALID_COMMAND: Invalid SMTP command from initiator (192.168.12.3:52419)
```

CBAC also detects a limited number of SMTP attack signatures. A signature in a SYSLOG message indicates a possible attack against the protected network, such as the detection of illegal SMTP commands in a packet. Whenever a signature is detected, the connection will be reset.

The Cisco Secure Integrated Software supports the following SMTP attack signatures.

Signature	Description
Mail: bad rcpt	Triggers on any mail message with a "pipe" (I) symbol in the recipient field.
Mail: bad from	Triggers on any mail message with a "pipe" (I) symbol in the "From:" field.
Mail: old attack	Triggers when "wiz" or "debug" commands are sent to the SMTP port.
Mail: decode	Triggers on any mail message with a ":decode@" in the header.
Majordomo	A bug in the Majordomo program will allow remote users to execute arbitrary commands at the privilege level of the server.

The following is a sample SMTP attack signature message:

```
02:04:55: %FW-4-TCP_MAJORDOMO_EXEC_BUG: Sig:3107:Majordomo Execute Attack - from
192.168.25.1 to 192.168.205.1:
```

Java Blocking Error Messages

CBAC detects and selectively blocks Java applets and notifies you when a Java applet has been blocked. Error messages such as the following may indicate that a Java applet has been blocked:

```
%FW-4-HTTP_JAVA_BLOCK: JAVA applet is blocked from (172.21.127.218:80) to
(172.16.57.30:44673).
```

FTP Error Messages

CBAC detects and prevents certain FTP attacks and notifies you when this occurs. Error messages such as the following may appear when CBAC detects these FTP attacks:

```
%FW-3-FTP_PRIV_PORT: Privileged port 1000 used in PORT command -- FTP client 10.0.0.1
FTP server 10.1.0.1
%FW-3-FTP_SESSION_NOT_AUTHENTICATED: Command issued before the session is authenticated
-- FTP client 10.0.0.1
%FW-3-FTP_NON_MATCHING_IP_ADDR: Non-matching address 172.19.148.154 used in PORT
 command -- FTP client 172.19.54.143  FTP server 172.16.127.242
```

Audit Trail Messages

CBAC provides audit trail messages to record details about inspected sessions. Audit trail information is configurable on a per-application basis using the CBAC inspection rules. To determine which protocol was inspected, use the responder's port number. The port number follows the responder's address. The following are sample audit trail messages:

```
%FW-6-SESS_AUDIT_TRAIL: tcp session initiator (192.168.1.13:33192) sent 22 bytes --
responder (192.168.129.11:25) sent 208 bytes
%FW-6-SESS_AUDIT_TRAIL: http session initiator (172.16.57.30:44673) sent 1599 bytes --
responder (172.21.127.218:80) sent 93124 bytes
```

Turning Off CBAC

You can turn off CBAC, with the **no ip inspect** global configuration command.

Note The **no ip inspect** command removes all CBAC configuration entries and resets all CBAC global timeouts and thresholds to the defaults. All existing sessions are deleted and their associated access lists removed.

In most situations, turning off CBAC has no negative security impact because CBAC creates "permit" access lists. Without CBAC configured, no "permit" access lists are maintained. Therefore, no derived traffic (returning traffic or traffic from the data channels) can go through the firewall. The exception is SMTP and Java blocking. With CBAC turned off, unacceptable SMTP commands or Java applets may go through the firewall.

CBAC Configuration Examples

The following sections provide CBAC configuration examples:

- Ethernet Interface Configuration Example
- ATM Interface Configuration Example
- Remote Office to ISP Configuration Example
- Remote Office to Branch Office Configuration Example
- Two-Interface Branch Office Configuration Example
- Multiple-Interface Branch Office Configuration Example

The first example develops a CBAC inspection rule for specific protocols and a supporting access control list (ACL). This example focuses how to configure CBAC; it does not provide a complete router configuration and does not describe other elements of the configuration.

The next example develops a CBAC inspection rule for sites that might have remote traffic through an ATM interface. This example further illustrates on how to configure CBAC and emphasizes the application of the configuration rule at the interface, whatever that interface might be. This example does not provide a complete router configuration and does not describe other elements of the configuration.

The remote-office examples also focus on the firewall configuration but do not provide detailed descriptions of other configuration elements, such as the Basic Rate Interface (BRI) and dialer interface configurations.

Other examples provide more complete firewall configurations, further illustrating ways in which to apply CBAC.

In each example, configuring protocol inspection using CBAC has four components:

- Defining an access list with the appropriate permissions.
- Applying the ACL at an interface where you want to control access.
- Defining an inspection rule that includes the protocol that you want to inspect.
- Applying the inspection rule at an interface where you want to inspect traffic.

Ethernet Interface Configuration Example

This example looks at each of these four components. For this example, CBAC is being configured to inspect RTSP and H.323 protocol traffic inbound from the protected network on a router with two Ethernet interfaces. Interface Ethernet1/0 is the protected network and interface Ethernet1/1 is the unprotected network. The security policy for the protected site uses access control lists (ACLs) to restrict inbound traffic on the unprotected interface to specific ICMP protocol traffic, denying inbound access for TCP and UDP protocol traffic. Inbound access for specific protocol traffic is provided through dynamic access lists, which are generated according to CBAC inspection rules.

ACL 100 denies TCP and UDP traffic from any source or destination while permitting specific ICMP protocol traffic. The final deny statement is not required, but is included for explicitness—the final entry in any ACL is an implicit denial of all IP protocol traffic.

```
Router(config)# access-list 100 deny tcp any any
Router(config)# access-list 100 deny udp any any
Router(config)# access-list 100 permit icmp any any echo-reply
Router(config)# access-list 100 permit icmp any any time-exceeded
Router(config)# access-list 100 permit icmp any any packet-too-big
Router(config)# access-list 100 permit icmp any any traceroute
Router(config)# access-list 100 permit icmp any any unreachable
Router(config)# access-list 100 deny ip any any
```

ACL 100 is applied inbound at interface Ethernet1/1 to block all access from the unprotected network to the protected network.

```
Router(config)# interface Ethernet1/1
Router(config-if)# ip access-group 100 in
```

An inspection rule is created for "hqusers" that covers two protocols: RTSP and H.323.

```
Router(config)# ip inspect name hqusers rtsp
Router(config)# ip inspect name hqusers h323
```

The inspection rule is applied inbound at interface Ethernet1/0 to inspect traffic from users on the protected network. When CBAC detects multimedia traffic from the protected network, CBAC creates dynamic entries in access list 100 to allow return traffic for multimedia sessions.

```
Router(config)# interface Ethernet1/0
Router(config-if)# ip inspect hqusers in
```

ATM Interface Configuration Example

In this example, CBAC inspection (firewall protection) is required against inbound traffic on an ATM interface. This example might apply to sites where local hosts require access to hosts or services on a remote network. The security policy for this site uses access control lists (ACLs) to restrict inbound traffic on the ATM interface to specific ICMP protocol traffic, denying inbound access for TCP and UDP protocol traffic. Inbound access for specific TCP and UDP protocol traffic is provided through dynamic access lists, which are generated according to CBAC inspection rules.

For information on how to select the interface on which to apply CBAC, refer to the "Picking an Interface: Internal or External" section.

Note For Frame Relay or ATM interfaces, you can apply CBAC inspection rules separately on each sub-interface, even though the sub-interfaces are physically connected through one interface.

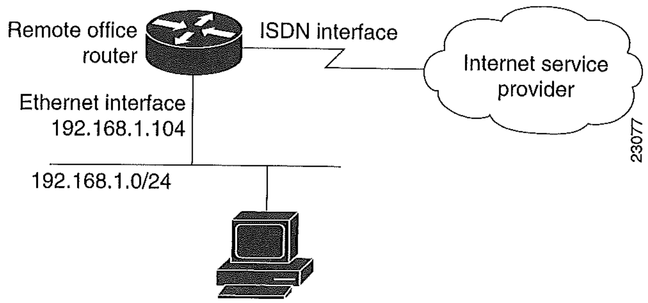
```
! -------------------------
! Create the Inspection Rule
! -------------------------
!
! Create the CBAC inspection rule "test", allowing inspection of the protocol traffic
! specified by the rule. This inspection rule sets the timeout value to 30 seconds for
! each protocol (except for RPC). The timeout value defines the maximum time that a
! connection for a given protocol can remain active without any traffic passing through
! the router. When these timeouts are reached, the dynamic ACLs that are inserted to
! permit the returning traffic are removed, and subsequent packets (possibly even valid
! ones) are not permitted.
ip inspect name test cuseeme timeout 30
ip inspect name test ftp timeout 30
ip inspect name test h323 timeout 30
ip inspect name test realaudio timeout 30
ip inspect name test rpc program-number 100000
ip inspect name test streamworks timeout 30
ip inspect name test vdolive timeout 30
!
! ------------------------------
! Create the Access Control List
! ------------------------------
!
```

```
! In this example, ACL 105 denies all TCP and UDP protocol traffic. ICMP traffic from
! subnet 192.168.1.0 is permitted to allow access for routing and control traffic.
! ACL 105 specifies that only the return traffic for protocols defined in the
! inspection rule is allow access through the interface where this rule is applied. The
! final deny statement is added for explicitness.
access-list 105 deny TCP any any
access-list 105 deny UDP any any
access-list 105 permit icmp any any echo-reply
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 time-exceeded
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 packet-too-big
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 traceroute
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 unreachable
access-list 105 deny ip any any
!
! ---------------------------------
! Apply the Inspection Rule and ACL
! ---------------------------------
!
! In this example, the inspection rule "test" is applied to traffic at interface ATM3/0
! for connections initiated in the outbound direction; that is, from hosts that are
! located on a local network. CBAC creates dynamic access list entries for traffic
! initiated by local hosts. These dynamic entries allow inbound (returning) traffic for
! that connection. ACL 105 is applied at interface ATM3/0 in the inbound direction to
! block traffic initiated from hosts on a remote network that is not part of an
! existing connection.
interface ATM3/0
   ip address 10.1.10.1 255.0.0.0
   ip access-group 105 in
   no ip directed-broadcast
   ip inspect test out
   no shutdown
   atm clock INTERNAL
   atm pvc 7 7 7 aal5snap
   map-group atm
```

Remote Office to ISP Configuration Example

This example describes one possible Cisco Secure Integrated Software configuration for a remote office router connected to an Internet service provider (ISP). In this configuration, the site security policy allows hosts on the local network to initiate traffic to the ISP while traffic inbound to the router from the ISP is blocked at the ISDN interface. Specific ICMP control message traffic is permitted through the firewall. No mail or Web services are available from the local network. Figure 15 illustrates this example.

Figure 15 **Remote Office to ISP Sample Configuration**



The firewall has two interfaces:

- An Ethernet interface connects to the internal protected network.

  Interface Ethernet0 has no ACL applied to it, meaning that all traffic initiated on the LAN is allowed access to the ISP. In this configuration example, Network Address Translation (NAT) is not turned on, and the addresses on interface Ethernet0 are reserved IP addresses. In a production environment, addresses on Ethernet0 either must be registered network addresses, or you must turn on NAT to hide these inside addresses from being visible on the Internet.

- An ISDN Basic Rate Interface (BRI) connects the router to the ISP. In this example, a dialer profile is used to control the BRI interface. This means that the ACL and CBAC inspection rules are applied at the dialer interface, not directly at the physical ISDN (BRI) interface using a dialer map.

```
! ---------------------------------------
! General Cisco Secure Integrated Software Guidelines
! ---------------------------------------
! The following global configuration entries illustrate good security practices.
enable secret 5 <elided>
no ip source-route
no cdp run
!
! --------------------------------
! Create the CBAC inspection rule
! --------------------------------
! Create the CBAC inspection rule STOP to allow inspection of the protocol traffic
! specified by the rule.
ip inspect name STOP tcp
ip inspect name STOP ftp
ip inspect name STOP smtp
ip inspect name STOP h323
ip inspect name STOP rcmd
!
! ---------------------------------
! Create Access Control List 105
! ---------------------------------
! ACL 105 denies all IP protocol traffic except for specific ICMP control traffic.
! This means that only the return traffic for protocols defined in the
! inspection rule and the specified ICMP traffic is allowed access through the
! interface where this rule is applied.
!
! Deny broadcast messages with a source address of 255.255.255.255; this helps to
! prevent broadcast attacks.
access-list 105 deny ip host 255.255.255.255 any
!
```

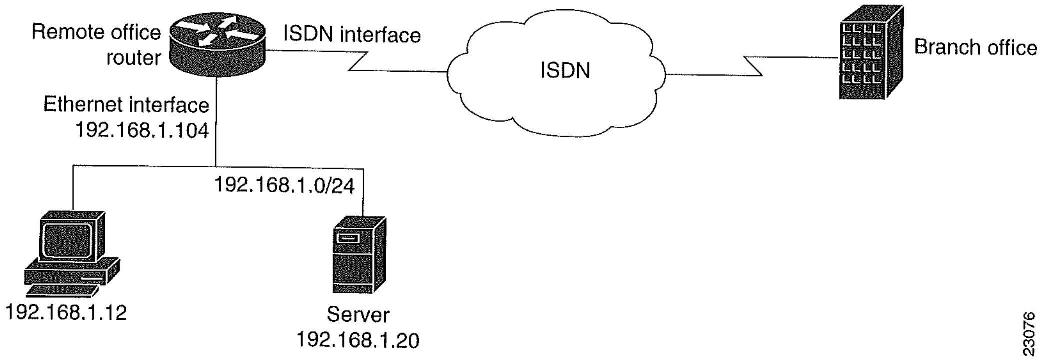
```
! Add anti-spoofing protection by denying traffic with a source address matching a host
! on the Ethernet interface.
acl 105 deny ip 192.168.1.0 0.0.0.255 any
!
! ICMP traffic is not inspected by CBAC. To control the type of ICMP traffic at the
! interface, add static access list entries. This example has the following ICMP
! requirements: outgoing ping commands require echo-reply messages to come back,
! outgoing traceroute commands require time-exceeded messages to come back, path MTU
! discovery requires "too-big" messages to come back, and incoming traceroute
! messages must be allowed. Additionally, permit all "unreachable" messages to come
! back; that is, if a router cannot forward or deliver a datagram, it sends an ICMP
! unreachable message back to the source and drops the datagram.
access-list 105 permit icmp any any echo-reply
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 time-exceeded
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 packet-too-big
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 traceroute
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 unreachable
!
! Final deny for explicitness. This entry is not required but helps complete the access
! list picture. By default, the final entry in any access list is an implicit deny of
! IP protocol traffic. This ensures that the firewall blocks any traffic not explicitly
! permitted by the access list.
access-list 105 deny ip any any
!
! ---------------------------------------------------------
! Configure the interface
! ---------------------------------------------------------
! In this example, no ACLs or inspection rules are applied at interface Ethernet0,
! meaning that all traffic on the local network is allowed to go out. This assumes a
! high-level of trust for the users on the local network.
interface Ethernet0
  ip address 192.168.1.104 255.255.255.0
!
no ip directed-broadcast
!
! This example uses a dialer profile, so the ACL and CBAC inspection rules are applied
! at the dialer interface, not the physical BRI interface. The dialer pool-member
! command is used to associate the physical interface with a dialer profile.
interface BRI0
  no ip address
  no ip directed-broadcast
  encapsulation ppp
  dialer pool-member 1
  isdn switch-type basic-5ess
!
! -----------------------------------------------------------------
! Create the dialer profile.
! -----------------------------------------------------------------
! Through the dialer profile, the ACL and CBAC inspection rules are
! applied to every pool member. In this example, the ACL is applied in, meaning that it
! applies to traffic inbound from the ISP. The CBAC inspection rule STOP is applied
! out, meaning that CBAC monitors the traffic through the interface and controls return
! traffic to the router for an existing connection.
interface Dialer0
  ip address negotiated
  ip access-group 105 in
  no ip directed-broadcast
  ip inspect STOP out
  encapsulation ppp
  dialer remote-name <ISP router>
  dialer idle-timeout 500
  dialer string <elided>
  dialer pool 1
```

```
   dialer-group 1
   ppp authentication callin
!
! ---------------------------------------------------------
! Additional entries
! ---------------------------------------------------------
! Configure the router to forward packets destined for an unrecognized subnet of
! a directly connected network.
ip classless
! Route traffic to the dialer interface.
ip route 0.0.0.0 0.0.0.0 Dialer0
! Include a dialer list protocol entry to specify the protocol that triggers dialing.
dialer-list 1 protocol ip permit
! Add a user name (name of the router your are configuring) and password for caller
! identification and password authentication with the ISP router.
username <router host name> password 5 <elided>
```

Remote Office to Branch Office Configuration Example

This example describes one possible Cisco Secure Integrated Software configuration for a remote office router connected to a branch office. In this configuration, the site security policy allows hosts on the local network to initiate traffic to the branch office. Mail or Web services are available from a server on the local network, and access to these services is available from the branch office. Traffic from the branch office, except for mail and Web traffic, is blocked at the outside interface. Specific ICMP control message traffic is permitted through the firewall. Figure 16 illustrates this example.

Figure 16 Remote Office to Branch Office Sample Configuration



The firewall has two interfaces:

- An Ethernet interface connects to the internal protected network.

  Interface Ethernet0 has no ACL applied to it, meaning that all traffic initiated from the LAN is allowed access through the firewall.

- An ISDN Basic Rate Interface (BRI) connects the router to the branch office. In this example, a dialer profile is used to control the BRI interface. This means that the ACL and CBAC inspection rules are applied at dialer interface, not directly at the physical ISDN (BRI) interface.

```
! --------------------------------------------
! General firewall configuration guidelines
! --------------------------------------------
! The following global configuration entries illustrate good security practices.
enable secret 5 <elided>
no ip source-route
no cdp run
!
! ---------------------------
! Create the Inspection Rule
! ---------------------------
! Create the CBAC inspection rule STOP to allow inspection of the specified protocol
! traffic. Create the inspection rule GO to allow inspection of SMTP traffic.
ip inspect name STOP tcp
ip inspect name STOP ftp
ip inspect name STOP smtp
ip inspect name STOP h323
ip inspect name GO smtp
!
! ------------------------------------------------------------
! Create Access Control Lists 106 and 51
! ------------------------------------------------------------
! ACL 106 permits mail and Web traffic from any host to the specified server. ACL 106
! denies all other ip protocol traffic except for specific ICMP control traffic.
! This means that only the return traffic for protocols defined in the
! inspection rule and the specified ICMP traffic is allowed access through the
! interface where this rule is applied.
!
! Deny broadcast messages with a source address of 255.255.255.255; this helps to
! prevent broadcast attacks.
access-list 106 deny ip host 255.255.255.255 any
!
! Add anti-spoofing protection by denying traffic with a source address matching a host
! on the Ethernet interface.
access-list 106 deny ip 192.168.1.0 0.0.0.255 any
!
! ICMP traffic is not inspected by CBAC. To control the type of ICMP traffic at the
! interface, add static access list entries. This example has the following ICMP
! requirements: outgoing ping commands require echo-reply messages to come back,
! outgoing traceroute commands require time-exceeded messages to come back, path MTU
! discovery requires "too-big" messages to come back, and incoming traceroute must be
! allowed. Additionally, permit all "unreachable" messages to come back; that is, if a
! router cannot forward or deliver a datagram, it sends an ICMP unreachable message
! back to the source and drops the datagram.
access-list 106 permit icmp any any echo-reply
access-list 106 permit icmp any 192.168.1.0 0.0.0.255 time-exceeded
access-list 106 permit icmp any 192.168.1.0 0.0.0.255 packet-too-big
access-list 106 permit icmp any 192.168.1.0 0.0.0.255 traceroute
access-list 106 permit icmp any 192.168.1.0 0.0.0.255 unreachable
!
! Permit mail and Web access to a specific server.
access-list 106 permit tcp any host 192.168.1.20 eq smtp
access-list 106 permit tcp any host 192.168.1.20 eq www
!
! Final deny for explicitness. This entry is not required but helps complete the access
! list picture. By default, the final entry in any access list is an implicit deny of
! IP protocol traffic. This ensures that the firewall blocks any traffic not explicitly
! permitted by the access list.
access-list 106 deny ip any any
!
! ------------------------------------------------------------
! Configure the interface.
! ------------------------------------------------------------
! In this example, no ACLs or inspection rules are applied at interface Ethernet0,
! meaning that all traffic on the local network is allowed to go out. This assumes a
! high-level of trust for the users on the local network.
interface Ethernet0
   ip address 192.168.1.104 255.255.255.0
   no ip directed-broadcast
!
! This example uses a dialer profile, so the ACL and CBAC inspection rules are applied
! at the dialer interface, not the physical BRI interface. The dialer pool-member
! command is used to associate the physical interface with a dialer profile.
interface BRI0
   no ip address
   no ip directed-broadcast
   encapsulation ppp
   dialer pool-member 1
   isdn switch-type basic-5ess
!
! ----------------------------------------------------------------
! Apply the ACL and CBAC inspection rules at the dialer interface.
! ----------------------------------------------------------------
! Through the dialer profile, the ACL and CBAC inspection rules are
! applied to every pool member. In this example, the ACL is applied in, meaning that it
! applies to traffic inbound from the branch office. The CBAC inspection rule STOP is
! applied out, meaning that CBAC monitors the traffic and controls return traffic to
! the router for an existing connection. The CBAC inspection rule GO is applied in,
! protecting against certain types of DoS attacks as described in this document. Note
! that the GO inspection rule does not control return traffic because there is no ACL
! blocking traffic in that direction; however, it does monitor the connections.
interface Dialer0
   ip address <ISDN interface address>
   ip access-group 106 in
   no ip directed-broadcast
   ip inspect STOP out
   ip inspect GO in
   encapsulation ppp
   dialer remote-name <branch office router>
   dialer idle-timeout 500
   dialer string <elided>
   dialer pool 1
   dialer-group 1
   ppp authentication
!
! -------------------------------------------------------
! Additional entries
! -------------------------------------------------------
! Configure the router to forward packets destined for an unrecognized subnet of
! a directly connected network.
ip classless
! Route traffic to the dialer interface.
ip route 0.0.0.0 0.0.0.0 Dialer0
! Include a dialer list protocol entry to specify the protocol that triggers dialing.
dialer-list 1 protocol ip permit
! Add a user name (name of the router your are configuring) and password for caller
! identification and password authentication with the ISP router.
username <router host name> password 5 <elided>
```

Two-Interface Branch Office Configuration Example

This sample configuration file describes a firewall configured with CBAC. The firewall is positioned between a protected field office's internal network and a WAN connection to the corporate headquarters. CBAC is configured on the firewall in order to protect the internal network from potential network threats coming from the WAN side.

The firewall has two interfaces configured:

- Interface Ethernet0 connects to the internal protected network
- Interface Serial0 connects to the WAN with Frame Relay

```
! ---------------------------------------------------------------------
! This first section contains some configuration that is not required for CBAC,
! but illustrates good security practices. Note that there are no
! services on the Ethernet side. Email is picked up via POP from a server on the
! corporate side.
! ---------------------------------------------------------------------
!
hostname user1-examplecorp-fr
!
boot system flash c1600-fw1600-l
enable secret 5 <elided>
!
username user1 password <elided>
ip subnet-zero
no ip source-route
ip domain-name example.com
ip name-server 172.19.2.132
ip name-server 198.92.30.32
!
!
! ---------------------------------------------------------------------
! The next section includes configuration required specifically for CBAC.
! ---------------------------------------------------------------------
!
! The following commands define the inspection rule "myfw", allowing
! the specified protocols to be inspected. Note that Java applets will be permitted
! according to access list 51, defined later in this configuration.
ip inspect name myfw cuseeme timeout 3600
ip inspect name myfw ftp timeout 3600
ip inspect name myfw http java-list 51 timeout 30
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
!
! The following interface configuration applies the "myfw" inspection rule to
! inbound traffic at Ethernet 0. Since this interface is on the internal network
! side of the firewall, traffic entering Ethernet 0 is actually
! exiting the internal network. Applying the inspection rule to this interface causes
! inbound traffic (which is exiting the network) to be inspected; return traffic will
! only be permitted back through the firewall if part of a session which began from
! within the network.
! Also note that access list 101 is applied to inbound traffic at Ethernet 0.
! (Traffic blocked by the access list will not be inspected.)
interface Ethernet0
description ExampleCorp Ethernet chez user1
ip address 172.19.139.1 255.255.255.248
ip broadcast-address 172.19.131.7
no ip directed-broadcast
no ip proxy-arp
ip inspect myfw in
ip access-group 101 in
no cdp enable
!
interface Serial0
description Frame Relay (Telco ID 22RTQQ062438-001) to ExampleCorp HQ
no ip address
```

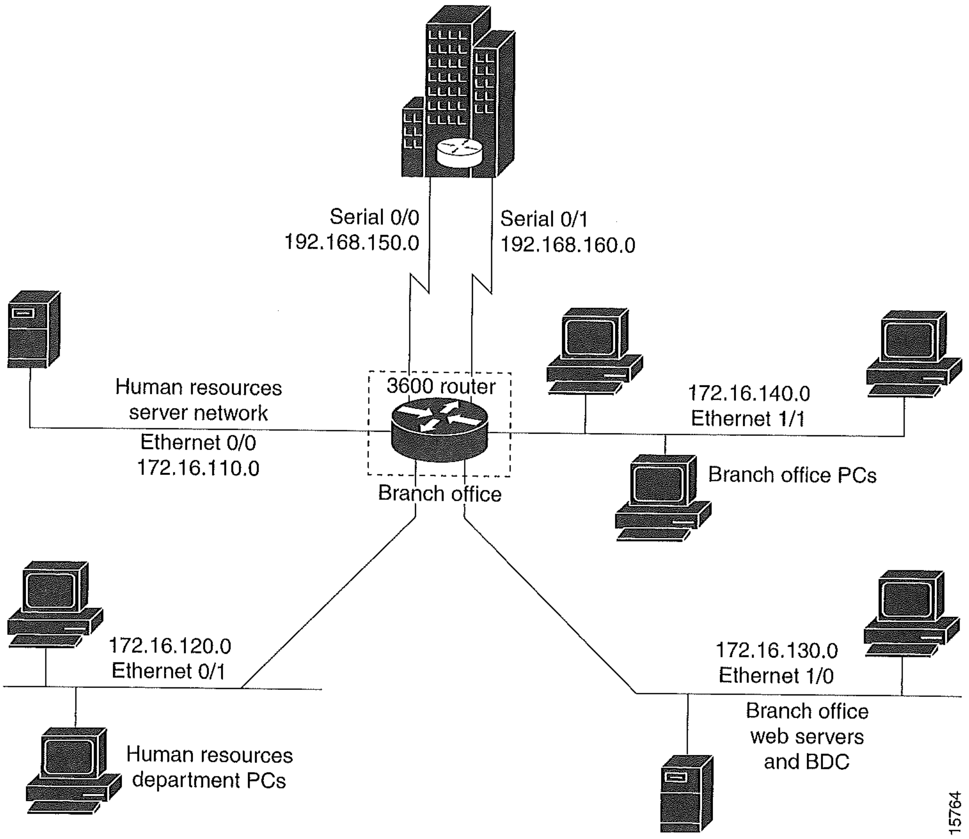
```
ip broadcast-address 0.0.0.0
encapsulation frame-relay IETF
no arp frame-relay
bandwidth 56
service-module 56k clock source line
service-module 56k network-type dds
frame-relay lmi-type ansi
!
! Note that the following interface configuration applies access list 111 to
! inbound traffic at the external serial interface. (Inbound traffic is
! entering the network.) When CBAC inspection occurs on traffic exiting the
! network, temporary openings will be added to access list 111 to allow returning
! traffic that is part of existing sessions.
!
interface Serial0.1 point-to-point
ip unnumbered Ethernet0
ip access-group 111 in
bandwidth 56
no cdp enable
frame-relay interface-dlci 16
!
ip classless
ip route 0.0.0.0 0.0.0.0 Serial0.1
!
! The following access list defines "friendly" and "hostile" sites for Java
! applet blocking. Because Java applet blocking is defined in the inspection
! rule "myfw" and references access list 51, applets will be actively denied
! if they are from any of the "deny" addresses and allowed only if they are from
! either of the two "permit" networks.
!
access-list 51 deny   172.19.1.203
access-list 51 deny   172.19.2.147
access-list 51 permit 172.18.0.0 0.1.255.255
access-list 51 permit 192.168.1.0 0.0.0.255
access-list 51 deny   any
!
! The following access list 101 is applied to interface Ethernet 0 above.
! This access list permits all traffic that should be CBAC inspected, and also
! provides anti-spoofing. The access list is deliberately set up to deny unknown
! IP protocols, because no such unknown protocols will be in legitimate use.
!
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
!
! The following access list 111 is applied to interface Serial 0.1 above.
! This access list filters traffic coming in from the external side. When
! CBAC inspection occurs, temporary openings will be added to the beginning of
! this access list to allow return traffic back into the internal network.
! This access list should restrict traffic that will be inspected by
! CBAC. (Remember that CBAC will open holes as necessary to permit returning traffic.)
! Comments precede each access list entry. These entries are not all specifically
! related to CBAC, but are created to provide general good security.
!
! Anti-spoofing.
access-list 111 deny   ip 172.19.139.0 0.0.0.7 any
! Sometimes EIGRP is run on the Frame Relay link. When you use an
! input access list, you have to explicitly allow even control traffic.
! This could be more restrictive, but there would have to be entries
! for the EIGRP multicast as well as for the office's own unicast address.
access-list 111 permit igrp any any
!
! These are the ICMP types actually used...
! administratively-prohibited is useful when you are trying to figure out why
! you cannot reach something you think you should be able to reach.
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 administratively-prohibited
!
! This allows network admins at headquarters to ping hosts at the field office:
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 echo
!
! This allows the field office to do outgoing pings
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 echo-reply
!
! Path MTU discovery requires too-big messages
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 packet-too-big
!
! Outgoing traceroute requires time-exceeded messages to come back
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 time-exceeded
!
! Incoming traceroute
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 traceroute
!
! Permits all unreachables because if you are trying to debug
! things from the remote office, you want to see them. If nobody ever did
! any debugging from the network, it would be more appropriate to permit only
! port unreachables or no unreachables at all.
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 unreachable
!
!
! These next two entries permit users on most ExampleCorp networks to Telnet to
! a host in the field office. This is for remote administration by the network admins.
access-list 111 permit tcp 172.18.0.0 0.1.255.255 host 172.19.139.1 eq telnet
access-list 111 permit tcp 192.168.1.0 0.0.0.255 host 172.19.139.1 eq telnet
!
! Final deny for explicitness
access-list 111 deny ip any any
!
no cdp run
snmp-server community <elided> RO
!
line con 0
exec-timeout 0 0
password <elided>
login local
line vty 0
exec-timeout 0 0
password <elided>
login local
length 35
line vty 1
exec-timeout 0 0
password 7 <elided>
login local
line vty 2
exec-timeout 0 0
password 7 <elided>
login local
line vty 3
exec-timeout 0 0
password 7 <elided>
login local
line vty 4
exec-timeout 0 0
password 7 <elided>
login local
!
scheduler interval 500
end
```

Multiple-Interface Branch Office Configuration Example

In this configuration example, a single Cisco 3600 series firewall router is positioned at a branch office. It has four internal networks and two WAN connections to the corporate headquarters. CBAC is configured on the firewall to protect two of the internal networks from potential network threats coming from the WAN side and from less secure internal networks. Anti-spoofing protection is added at each interface with client systems. Figure 17 illustrates this configuration.

Note This example shows a moderately high level of trust by the administrators toward the expected users. Additional protection could be added to this configuration for a situation in a lower level of trust. That configuration would include ICMP filtering statements, significantly more protocol and address control through the use of more restrictive access control lists, and anti-spoofing applied everywhere. This configuration does not contain those additional restrictions because that would detract from the CBAC example.

Figure 17 Sample Cisco Secure Integrated Software Application Environment



The branch office has this sample network configuration:

- Ethernet interface 0/0 supports the Human Resources department servers. This network includes an email (SMTP and POP3) host and a Windows NT server. The Windows NT server is the Primary Domain Controller (PDC) for the Human Resources domain and has a trust relationship with the rest of the company; however, it contains applications and databases that must not be accessed by the rest of the company or the other groups in the branch office. The devices on this LAN are accessible only by users in the Human Resources department on Ethernet interface 0/1. The Mail server must be able to send and receive email (through SMTP sessions) with all other devices. The Windows 95 machines can use this machine as their email server (for sending email through SMTP sessions) and as a repository for accumulating email that they can then download through POP3 sessions. No one else in the company is allowed to form POP3 sessions to any machine on this LAN.
- Ethernet interface 0/1 supports the Windows 95 computers in the Human Resources department. These users must have access to the Human Resources mail servers located on Ethernet interface 0/0 as well as access to the rest of the company. Access to the Windows NT server resources are controlled through the Windows NT permissions assigned to each user in the Windows NT domain.
- Ethernet interface 1/0 supports the branch office web servers, which can be accessed by everyone in the company. These servers use TCP ports 80 (HTTP) and 443 (SHTTP) for inbound Web access. This network also includes a backup domain controller (BDC) for the overall domain that is also used as file, print, and service server.

  Ethernet interface 1/1 supports all users who are not in the Human Resources department. These users have no access to the Human Resources department servers, but they can access the other network interfaces and the serial interfaces for WAN connectivity. Serial interface 0/0 and 0/1 connect to the WAN with T1 links (links to corporate headquarters). In this sample configuration, the Domain Name System (DNS) servers are located somewhere within the rest of the company.

  Additionally, network management (SNMP) and Telnet sessions are limited to the management network (192.168.55.0), which is located somewhere within the rest of the company across the serial interface.

```
! -----------------------------------------------------------------
! This first section contains some configuration that is not required
! for CBAC, but illustrates good security practices.
! -----------------------------------------------------------------
! Add this line to get timestamps on the syslog messages.
service timestamps log datetime localtime show-timezone
!
hostname Router1
!
boot system flash c3600-fw3600-1
!
! Configure AAA user authentication.
aaa new-model
aaa authentication login lista group tacacs+ enable
!
enable secret 5 <elided>
ip subnet-zero
!
! Disable source routing to help prevent spoofing.
no ip source-route
!
! Set up the domain name and server IP addresses.
ip domain-name example.com
ip name-server 192.168.55.132
ip name-server 192.168.27.32
!
```

```
! The audit-trail command enables the delivery of specific CBAC messages
! through the syslog notification process.
ip inspect audit-trail
!
! Establish the time-out values for DNS queries. When this idle-timer expires,
! the dynamic ACL entries that were created to permit the reply to a DNS request
! will be removed and any subsequent packets will be denied.
ip inspect dns-timeout 10
!
! ---------------------------------------------------------------------------------
! The next section includes configuration statements required specifically for CBAC.
! ---------------------------------------------------------------------------------
! Define the CBAC inspection rule "inspect1", allowing the specified protocols to be
! inspected. The first rule enables SMTP specific inspection. SMTP inspection causes
! the exchange of the SMTP session to be inspected for illegal commands. Any packets
! with illegal commands are dropped, and the SMTP session will hang and eventually
! time out.
ip inspect name inspect1 smtp timeout 30
!
! In the next two lines of inspect1, define the maximum time that each of the UDP and
! TCP sessions are allowed to continue without any traffic passing
! through the router. When these timeouts are reached, the dynamic ACLs that
! are inserted to permit the returning traffic are removed and subsequent packets
! (possibly even valid ones) will not be permitted.
ip inspect name inspect1 udp timeout 30
ip inspect name inspect1 tcp timeout 30
!
! Define the CBAC inspection rule "inspect2", allowing the specified protocols to be
! inspected. These rules are similar to those used in the inspection rule "inspect1,"
! except that on the interfaces where this rule is applied, SMTP sessions are not
! expected to go through; therefore, the SMTP rule element is not applied here.
ip inspect name inspect2 udp timeout 30
ip inspect name inspect2 tcp timeout 30
!
! ----------------------------------------------------------------------
! The next section shows the Ethernet interface configuration statements for each
! interface, including access lists and inspections rules.
! ----------------------------------------------------------------------
! Apply the "inspect1" inspection rule to sessions that are initiated in the outbound
! direction (toward the LAN) at Ethernet interface 0/0. All packets in these sessions
! will be inspected by CBAC. Provided that network traffic passes the Access Control
! List (ACL) restrictions, traffic is then inspected by CBAC for access through the
! Cisco Secure Integrated Software. Traffic blocked by the access list is not inspected
! by CBAC. Access list 110 is applied to outbound traffic on this interface.
interface Ethernet0/0
   description HR_Server Ethernet
   ip address 172.16.110.1 255.255.255.0
   ip access-group 110 out
   no ip directed-broadcast
   no ip proxy-arp
   ip inspect inspect1 out
   no cdp enable
!
! Apply access list 120 to inbound traffic on Ethernet interface 0/1.
! Applying access list 120 to inbound traffic provides anti-spoofing on this interface
! by dropping traffic with a source address matching the IP address on a network other
! than Ethernet 0/1. The IP helper address lists the IP address of the DHCP server on
! Ethernet interface 1/0.
interface Ethernet0/1
   description HR_client Ethernet
   ip address 172.16.120.1 255.255.255.0
   ip access-group 120 in
   ip helper-address 172.16.130.66
   no ip directed-broadcast
   no ip proxy-arp
   no cdp enable
!
! Apply the "inspect2" inspection rule to sessions that are initiated in the outbound
! direction (toward the LAN) at Ethernet interface 1/0. Provided that network traffic
! passes the Access Control List (ACL) restrictions, traffic is then inspected by CBAC
! through the Cisco Secure Integrated Software. Traffic blocked by the access list is
! not inspected
! by CBAC. Access list 130 is applied to outbound traffic on this interface.
interface Ethernet1/0
   description Web_server Ethernet
   ip address 172.16.130.1 255.255.255.0
   ip access-group 130 out
   no ip directed-broadcast
   no ip proxy-arp
   ip inspect inspect2 out
   no cdp enable
!
! Apply access list 140 to inbound traffic at Ethernet interface 1/1. This
! provides anti-spoofing on the interface by dropping traffic with a source address
! matching the IP address of a network other than Ethernet 1/1. The IP helper address
! lists the IP address of the DHCP server on Ethernet interface 1/0.
interface Ethernet1/1
   description Everyone_else Ethernet
   ip address 172.16.140.1 255.255.255.0
   ip access-group 140 in
   ip helper-address 172.16.130.66
   no ip directed-broadcast

   no ip proxy-arp
   no cdp enable
!
! -----------------------------------------------------------------------
! The next section configures the serial interfaces, including access lists.
! -----------------------------------------------------------------------
! Apply access list 150 to Serial interfaces 0/0. This provides anti-spoofing on the
! serial interface by dropping traffic with a source address matching the IP address
! of a host on Ethernet interface 0/0, 0/1, 1/0, or 1/1.
interface Serial0/0
   description T1 to HQ
   ip address 192.168.150.1 255.255.255.0
   ip access-group 150 in
   bandwidth 1544
!
interface Serial1/1
   description T1 to HQ
   ip address 192.168.160.1 255.255.255.0
   ip access-group 150 in
   bandwidth 1544
!
! ------------------------------
! Configure routing information.
! ------------------------------
router igrp 109
network 172.16.0.0
network 192.168.150.0
network 192.168.160.0
!
! Define protocol forwarding on the firewall. When you turn on a related command,
! ip helper-address, you forward every IP broadcast in the ip forward protocol
! command list, including several which are on by default: TFTP (port 69),
! DNS (port 53), Time service (port 37), NetBIOS Name Server (port 137),
! NetBIOS Datagram Server (port 138), BOOTP client and server datagrams
! (ports 67 and 68), and TACACS service (port 49). One common
! application that requires helper addresses is Dynamic Host Configuration
! Protocol (DHCP). DHCP information is carried inside of BOOTP packets. The
! "no ip forward protocol" statements turn off forwarding for the specified protocols.
no ip forward-protocol udp netbios-ns
no ip forward-protocol udp netbios-dgm
no ip forward-protocol udp tacacs
no ip forward-protocol udp tftp
ip forward-protocol udp bootpc
!
! Add this line to establish where router SYSLOG messages are sent. This includes the
! CBAC messages.
logging 192.168.55.131
!
! -----------------------------------------------------------
! Define the configuration of each access list.
! -----------------------------------------------------------
! Defines Telnet controls in access list 12.
access-list 12 permit 192.168.55.0 0.0.0.255
!
! Defines SNMP controls in access list 13.
access-list 13 permit 192.168.55.12
access-list 13 permit 192.168.55.19
!
! Access list 110 permits TCP and UDP protocol traffic for specific ports and with a
! source address on Ethernet interface 0/1. The access list denies IP protocol traffic
! with any other source and destination address. The access list permits ICMP access
! for any source and destination address. Access list 110 is deliberately set up to
! deny unknown IP protocols because no such unknown protocols will be in legitimate
! use. Access list 110 is applied to outbound traffic at Ethernet interface 0/0. In ACL
! 110, network traffic is being allowed access to the ports on any server on the HR
! server network. In less trusted environments, this can be a security problem;
! however, you can limit access more severely by specifying specific destination
! addresses in the ACL statements.
access-list 110 permit tcp 172.16.120.0 0.0.0.255 any eq smtp
access-list 110 permit tcp 172.16.120.0 0.0.0.255 any eq pop3
access-list 110 permit tcp 172.16.120.0 0.0.0.255 any eq 110
access-list 110 permit udp any any eq 137
access-list 110 permit udp any any eq 138
access-list 110 permit udp any any eq 139
access-list 110 permit icmp any any
access-list 110 deny ip any any!
!
! Access-list 120 permits TCP, UDP, and ICMP protocol traffic with a source address
! on Ethernet interface 0/1, but denies all other IP protocol traffic. Access list
! 120 is applied to inbound traffic on Ethernet interface 0/1.
access-list 120 permit tcp 172.16.120.0 0.0.0.255 any
access-list 120 permit udp 172.16.120.0 0.0.0.255 any
access-list 120 permit icmp 172.16.120.0 0.0.0.255 any
access-list 120 deny ip any any
!
! Access list 130 permits TCP, UDP, and ICMP protocol traffic for specific ports and
! with any source and destination address. It opens access to the web server and to
! all NBT services to the rest of the company, which can be controlled through the
! trust relations on the Windows NT servers. The bootpc entry permits access to the
! DHCP server. Access list 130 denies all other IP protocol traffic. Access list 130 is
! applied to outbound traffic at Ethernet interface 1/0.
access-list 130 permit tcp any any eq www
access-list 130 permit tcp any any eq 443
access-list 130 permit tcp any any eq 110
access-list 130 permit udp any any eq 137
access-list 130 permit udp any any eq 138
access-list 130 permit udp any any eq 139
access-list 130 permit udp any any eq bootpc
access-list 130 permit icmp any any
access-list 130 deny ip any any
!
! Access list 140 permits TCP, UDP, and ICMP protocol traffic with a source address on
! Ethernet interface 1/1, and it denies all other IP protocol traffic. Access list 140
! is applied to inbound traffic at Ethernet interface 1/1.
access-list 140 permit tcp 172.16.140.0 0.0.0.255 any
access-list 140 permit udp 172.16.140.0 0.0.0.255 any
access-list 140 permit icmp 172.16.140.0 0.0.0.255 any
access-list 140 deny ip any any
!
! Access list 150 denies IP protocol traffic with a source address on Ethernet
! interfaces 0/0, 0/1, 1/0, and 1/1, and it permits IP protocol traffic with any other
! source and destination address. Access list 150 is applied to inbound traffic
! on each of the serial interfaces.
access-list 150 deny ip 172.16.110.0 0.0.0.255 any
access-list 150 deny ip 172.16.120.0 0.0.0.255 any
access-list 150 deny ip 172.16.130.0 0.0.0.255 any
access-list 150 deny ip 172.16.140.0 0.0.0.255 any
access-list 150 permit ip any any
!
! Disable Cisco Discovery Protocol.
no cdp run
!
snmp-server community <elided> ro 13
tacacs-server host 192.168.55.2
tacacs-server key <elided>
!
! ---------------------------------------------------------------------------------
! Configures the router console port and the virtual terminal line interfaces,
! including AAA authentication at login. Authentication is required for users defined
! in "lista." Access-class 12 is applied on each line, restricting Telnet access to
! connections with a source address on the network management network.
! ---------------------------------------------------------------------------------
line console 0
exec-timeout 3 00
login authentication lista
line aux 0
exec-timeout 3 00
login authentication lista
line vty 0
   exec-timeout 1 30
   login authentication lista
   access-class 12 in
line vty 1
   exec-timeout 1 30
   login authentication lista
   access-class 12 in
line vty 2
   exec-timeout 1 30
   login authentication lista
   access-class 12 in
line vty 3
   exec-timeout 1 30
   login authentication lista
   access-class 12 in
line vty 4
   exec-timeout 1 30
   login authentication lista
   access-class 12 in
!
end
```

Configuring Cisco Secure Integrated Software Intrusion Detection System

This chapter describes the Cisco Secure Integrated Software (IS) Intrusion Detection System (IDS) feature. Intrusion detection systems provide a level of protection beyond the firewall by protecting the network from internal and external attacks and threats. Cisco Secure IS IDS technology enhances perimeter firewall protection by taking appropriate action on packets and flows that violate the security policy or represent malicious network activity.

For a complete description of the Cisco Secure IS IDS commands in this chapter, refer to the "Cisco Secure Integrated Software IDS Commands" chapter of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

In This Chapter

This chapter has the following sections:

- Feature Overview
- Cisco Secure IS IDS Configuration Task List
- Monitoring and Maintaining Cisco Secure IS IDS
- Cisco Secure IS IDS Configuration Examples

Feature Overview

The Cisco Secure Integrated Software IDS supports intrusion detection technology for midrange and high-end router platforms with firewall support. It is ideal for any network perimeter, and especially for locations in which a router is being deployed and additional security between network segments is required. It also can protect intranet and extranet connections where additional security is mandated, and branch-office sites connecting to the corporate office or Internet.

The Cisco Secure IS IDS feature identifies 59 of the most common attacks using "signatures" to detect patterns of misuse in network traffic. The intrusion-detection signatures included in the Cisco Secure Integrated Software were chosen from a broad cross-section of intrusion-detection signatures. The signatures represent severe breaches of security and the most common network attacks and information-gathering scans. For a description of Cisco Secure IS IDS signatures, refer to the "Cisco Secure IS IDS Signature List" section.

The Cisco Secure IS IDS acts as an in-line intrusion detection sensor, watching packets and sessions as they flow through the router, scanning each to match any of the IDS signatures. When it detects suspicious activity, it responds before network security can be compromised and logs the event through Cisco IOS syslog or the Cisco Secure Intrusion Detection System (Cisco Secure IDS, formerly known as Net Ranger) Post Office Protoco The network administrator can configure the IDS system to choose the appropriate response to various threats. When packets in a session match a signature, the IDS system can be configured to take these actions:

- Send an alarm to a syslog server or a Cisco Secure IDS Director (centralized management interface)
- Drop the packet
- Reset the TCP connection

Cisco developed its Cisco IOS software-based intrusion-detection capabilities in Cisco Secure Integrated Software with flexibility in mind, so that individual signatures could be disabled in case of false positives. Also, while it is preferable to enable both the firewall and intrusion detection features of the CBAC security engine to support a network security policy, each of these features may be enabled independently and on different router interfaces. Cisco IOS software-based intrusion detection is part of the Cisco Secure Integrated Software.

This section has the following sections:

- Compatibility with Cisco Secure Intrusion Detection
- Functional Description
- When to Use Cisco Secure IS IDS
- Memory and Performance Impact
- Cisco Secure IS IDS Signature List

Compatibility with Cisco Secure Intrusion Detection

Cisco Secure Integrated Software is compatible with the Cisco Secure Intrusion Detection System (formally known as NetRanger). The Cisco Secure IDS is an enterprise-scale, real-time, intrusion detection system designed to detect, report, and terminate unauthorized activity throughout a network.

The Cisco Secure IDS consists of three components:

- Sensor
- Director
- Post Office

Cisco Secure IDS Sensors, which are high-speed network appliances, analyze the content and context of individual packets to determine if traffic is authorized. If a network's data stream exhibits unauthorized or suspicious activity, such as a SATAN attack, a ping sweep, or the transmission of a secret research project code word, Cisco Secure IDS Sensors can detect the policy violation in real time, forward alarms to a Cisco Secure IDS Director management console, and remove the offender from the network.

The Cisco Secure IDS Director is a high-performance, software-based management system that centrally monitors the activity of multiple Cisco Secure IDS Sensors located on local or remote network segments.

The Cisco Secure IDS Post Office is the communication backbone that allows Cisco Secure IDS services and hosts to communicate with each other. All communication is supported by a proprietary, connection-based protocol that can switch between alternate routes to maintain point-to-point connections.

Cisco Secure IDS customers can deploy the Cisco Secure IS IDS signatures to complement their existing IDS systems. This allows an IDS to be deployed to areas that may not be capable of supporting a Cisco Secure IDS Sensor. Cisco Secure IS IDS signatures can be deployed alongside or independently of other Cisco Secure Integrated Software features.

The Cisco Secure IS IDS can be added to the Cisco Secure IDS Director screen as an icon to provide a consistent view of all intrusion detection sensors throughout a network. The Cisco Secure Integrated Software intrusion detection capabilities have an enhanced reporting mechanism that permits logging to the Cisco Secure IDS Director console in addition to Cisco IOS syslog.

For additional information about Cisco Secure IDS (NetRanger), refer to the *NetRanger User Guide*.

Functional Description

The Cisco Secure IS IDS acts as an in-line intrusion detection sensor, watching packets as they traverse the router's interfaces and acting upon them in a definable fashion. When a packet, or a number of packets in a session, match a signature, the Cisco Secure Integrated Software IDS may perform the following configurable actions:

- Alarm—Sends an alarm to a syslog server or Cisco Secure IDS Director
- Drop—Drops the packet
- Reset—Resets the TCP connection

The following describes the packet auditing process with Cisco Secure IS IDS:

- You create an audit rule, which specifies the signatures that should be applied to packet traffic and the actions to take when a match is found. An audit rule can apply informational and attack signatures to network packets. The signature list can have just one signature, all signatures, or any number of signatures in between. Signatures can be disabled in case of false positives or the needs of the network environment.
- You apply the audit rule to an interface on the router, specifying a traffic direction (*in* or *out*).
- If the audit rule is applied to the *in* direction of the interface, packets passing through the interface are audited before the inbound ACL has a chance to discard them. This allows an administrator to be alerted if an attack or information-gathering activity is underway even if the router would normally reject the activity.
- If the audit rule is applied to the *out* direction on the interface, packets are audited after they enter the router through another interface. In this case, the inbound ACL of the other interface may discard packets before they are audited. This may result in the loss of Cisco Secure IS IDS alarms even though the attack or information-gathering activity was thwarted.
- Packets going through the interface that match the audit rule are audited by a series of modules, starting with IP; then either ICMP, TCP, or UDP (as appropriate); and finally, the Application level.
- If a signature match is found in a module, then the following user-configured action(s) occur:
    - If the action is **alarm**, then the module completes its audit, sends an alarm, and passes the packet to the next module.
    - If the action is **drop**, then the packet is dropped from the module, discarded, and not sent to the next module.
    - If the action is **reset**, then the packets are forwarded to the next module, and packets with the reset flag set are sent to both participants of the session, if the session is TCP.

Note It is recommended that you use the **drop** and **reset** actions together.

If there are multiple signature matches in a module, only the first match fires an action. Additional matches in other modules fire additional alarms, but only one per module.

Note This process is different than on the Cisco Secure IDS Sensor appliance, which identifies all signature matches for each packet.

When to Use Cisco Secure IS IDS

Cisco Secure Integrated Software IDS capabilities are ideal for providing additional visibility at intranet, extranet, and branch-office Internet perimeters. Network administrators enjoy more robust protection against attacks on the network and can automatically respond to threats from internal or external hosts.

The Cisco Secure Integrated Software with intrusion detection is intended to satisfy the security goals of all of our customers, and is particularly appropriate for the following scenarios:

- Enterprise customers that are interested in a cost-effective method of extending their perimeter security across all network boundaries, specifically branch-office, intranet, and extranet perimeters.
- Small and medium-sized businesses that are looking for a cost-effective router that has an integrated firewall with intrusion-detection capabilities.
- Service provider customers that want to set up managed services, providing firewalling and intrusion detection to their customers, all housed within the necessary function of a router.

Memory and Performance Impact

The performance impact of intrusion detection will depend on the configuration of the signatures, the level of traffic on the router, the router platform, and other individual features enabled on the router such as encryption, source route bridging, and so on. Enabling or disabling individual signatures will not alter performance significantly, however, signatures that are configured to use Access Control Lists will have a significant performance impact.

Because this router is being used as a security device, no packet will be allowed to bypass the security mechanisms. The IDS process in the Cisco Secure Integrated Software router sits directly in the packet path and thus will search each packet for signature matches. In some cases, the entire packet will need to be searched, and state information and even application state and awareness must be maintained by the router.

For auditing atomic signatures, there is no traffic-dependent memory requirement. For auditing compound signatures, CBAC allocates memory to maintain the state of each session for each connection. Memory is also allocated for the configuration database and for internal caching.

Cisco Secure IS IDS Signature List

The following is a complete list of Cisco Secure IS IDS signatures. A signature detects patterns of misuse in network traffic. In Cisco Secure IS IDS, signatures are categorized into four types:

- Info Atomic
- Info Compound
- Attack Atomic
- Attack Compound

An info signature detects information-gathering activity, such as a port sweep.

An attack signature detects attacks attempted into the protected network, such as denial-of-service attempts or the execution of illegal commands during an FTP session.

Info and attack signatures can be either atomic or compound signatures. Atomic signatures can detect patterns as simple as an attempt to access a specific port on a specific host. Compound signatures can detect complex patterns, such as a sequence of operations distributed across multiple hosts over an arbitrary period of time.

The intrusion-detection signatures included in the Cisco Secure IS were chosen from a broad cross-section of intrusion-detection signatures as representative of the most common network attacks and information-gathering scans that are not commonly found in an operational network.

The following signatures are listed in numerical order by their signature number in the Cisco Secure IDS Network Security Database. After each signature's name is an indication of the type of signature (info or attack, atomic, or compound).

Note Atomic signatures marked with an asterisk (Atomic*) are allocated memory for session states by CBAC.

1000 IP options-Bad Option List (Info, Atomic)

Triggers on receipt of an IP datagram where the list of IP options in the IP datagram header is incomplete or malformed. The IP options list contains one or more options that perform various network management or debugging tasks.

1001 IP options-Record Packet Route (Info, Atomic)

Triggers on receipt of an IP datagram where the IP option list for the datagram includes option 7 (Record Packet Route).

1002 IP options-Timestamp (Info, Atomic)

Triggers on receipt of an IP datagram where the IP option list for the datagram includes option 4 (Timestamp).

1003 IP options-Provide s,c,h,tcc (Info, Atomic)

Triggers on receipt of an IP datagram where the IP option list for the datagram includes option 2 (Security options).

1004 IP options-Loose Source Route (Info, Atomic)

Triggers on receipt of an IP datagram where the IP option list for the datagram includes option 3 (Loose Source Route).

1005 IP options-SATNET ID (Info, Atomic)

Triggers on receipt of an IP datagram where the IP option list for the datagram includes option 8 (SATNET stream identifier).

1006 IP options-Strict Source Route (Info, Atomic)

Triggers on receipt of an IP datagram in which the IP option list for the datagram includes option 2 (Strict Source Routing).

1100 IP Fragment Attack (Attack, Atomic)

Triggers when any IP datagram is received with the "more fragments" flag set to 1 or if there is an offset indicated in the offset field.

1101 Unknown IP Protocol (Attack, Atomic)

Triggers when an IP datagram is received with the protocol field set to 101 or greater. These protocol types are undefined or reserved and should not be used.

1102 Impossible IP Packet (Attack, Atomic)

This triggers when an IP packet arrives with source equal to destination address. This signature will catch the so-called Land Attack.

2000 ICMP Echo Reply (Info, Atomic)

Triggers when a IP datagram is received with the "protocol" field in the IP header set to 1 (ICMP) and the "type" field in the ICMP header set to 0 (Echo Reply).

2001 ICMP Host Unreachable (Info, Atomic)

Triggers when an IP datagram is received with the "protocol" field in the IP header set to 1 (ICMP) and the "type" field in the ICMP header set to 3 (Host Unreachable).

2002 ICMP Source Quench (Info, Atomic)

Triggers when an IP datagram is received with the "protocol" field in the IP header set to 1 (ICMP) and the "type" field in the ICMP header set to 4 (Source Quench).

2003 ICMP Redirect (Info, Atomic)

Triggers when an IP datagram is received with the "protocol" field in the IP header set to 1 (ICMP) and the "type" field in the ICMP header set to 5 (Redirect).

2004 ICMP Echo Request (Info, Atomic)

Triggers when an IP datagram is received with the "protocol" field in the IP header set to 1 (ICMP) and the "type" field in the ICMP header set to 8 (Echo Request).

2005 ICMP Time Exceeded for a Datagram (Info, Atomic)

Triggers when an IP datagram is received with the "protocol" field in the IP header set to 1 (ICMP) and the "type" field in the ICMP header set to 11(Time Exceeded for a Datagram).

2006 ICMP Parameter Problem on Datagram (Info, Atomic)

Triggers when an IP datagram is received with the "protocol" field in the IP header set to 1 (ICMP) and the "type" field in the ICMP header set to 12 (Parameter Problem on Datagram).

2007 ICMP Timestamp Request (Info, Atomic)

Triggers when an IP datagram is received with the "protocol" field in the IP header set to 1 (ICMP) and the "type" field in the ICMP header set to 13 (Timestamp Request).

2008 ICMP Timestamp Reply (Info, Atomic)

Triggers when an IP datagram is received with the "protocol" field in the IP header set to 1 (ICMP) and the "type" field in the ICMP header set to 14 (Timestamp Reply).

2009 ICMP Information Request (Info, Atomic)

Triggers when an IP datagram is received with the "protocol" field in the IP header set to 1 (ICMP) and the "type" field in the ICMP header set to 15 (Information Request).

2010 ICMP Information Reply (Info, Atomic)

Triggers when an IP datagram is received with the "protocol" field in the IP header set to 1 (ICMP) and the "type" field in the ICMP header set to 16 (ICMP Information Reply).

2011 ICMP Address Mask Request (Info, Atomic)

Triggers when an IP datagram is received with the "protocol" field in the IP header set to 1 (ICMP) and the "type" field in the ICMP header set to 17 (Address Mask Request).

2012 ICMP Address Mask Reply (Info, Atomic)

Triggers when an IP datagram is received with the "protocol" field in the IP header set to 1 (ICMP) and the "type" field in the ICMP header set to 18 (Address Mask Reply).

2150 Fragmented ICMP Traffic (Attack, Atomic)

Triggers when an IP datagram is received with the protocol field in the IP header set to 1 (ICMP) and either the more fragments flag is set to 1 (ICMP) or there is an offset indicated in the offset field.

2151 Large ICMP Traffic (Attack, Atomic)

Triggers when an IP datagram is received with the protocol field in the IP header set to 1 (ICMP) and the IP length is greater than 1024.

2154 Ping of Death Attack (Attack, Atomic)

Triggers when an IP datagram is received with the protocol field in the IP header set to 1 (ICMP), the Last Fragment bit is set, and

```
( IP offset * 8 ) + (IP data length)  > 65535
```

In other words, the IP offset (which represents the starting position of this fragment in the original packet, and which is in 8-byte units) plus the rest of the packet is greater than the maximum size for an IP packet.

3040 TCP - no bits set in flags (Attack, Atomic)

Triggers when a TCP packet is received with no bits set in the flags field.

3041 TCP - SYN and FIN bits set (Attack, Atomic)

Triggers when a TCP packet is received with both the SYN and FIN bits set in the flag field.

3042 TCP - FIN bit with no ACK bit in flags (Attack, Atomic)

Triggers when a TCP packet is received with the FIN bit set but with no ACK bit set in the flags field.

3050 Half-open SYN Attack/SYN Flood (Attack, Compound)

Triggers when multiple TCP sessions have been improperly initiated on any of several well-known service ports. Detection of this signature is currently limited to FTP, Telnet, HTTP, and e-mail servers (TCP ports 21, 23, 80, and 25 respectively).

3100 Smail Attack (Attack, Compound)

Triggers on the very common "smail" attack against SMTP-compliant e-mail servers (frequently sendmail).

3101 Sendmail Invalid Recipient (Attack, Compound)
Triggers on any mail message with a "pipe" (|) symbol in the recipient field.

3102 Sendmail Invalid Sender (Attack, Compound)
Triggers on any mail message with a "pipe" (|) symbol in the "From:" field.

3103 Sendmail Reconnaissance (Attack, Compound)
Triggers when "expn" or "vrfy" commands are issued to the SMTP port.

3104 Archaic Sendmail Attacks (Attack, Compound)
Triggers when "wiz" or "debug" commands are issued to the SMTP port.

3105 Sendmail Decode Alias (Attack, Compound)
Triggers on any mail message with ": decode@" in the header.

3106 Mail Spam (Attack, Compound)
Counts number of Rcpt to: lines in a single mail message and alarms after a user-definable maximum has been exceeded (default is 250).

3107 Majordomo Execute Attack (Attack, Compound)
A bug in the Majordomo program will allow remote users to execute arbitrary commands at the privilege level of the server.

3150 FTP Remote Command Execution (Attack, Compound)
Triggers when someone tries to execute the FTP SITE command.

3151 FTP SYST Command Attempt (Info, Compound)
Triggers when someone tries to execute the FTP SYST command.

3152 FTP CWD ~root (Attack, Compound)
Triggers when someone tries to execute the CWD ~root command.

3153 FTP Improper Address Specified (Attack, Atomic*)
Triggers if a port command is issued with an address that is not the same as the requesting host.

3154 FTP Improper Port Specified (Attack, Atomic*)
Triggers if a port command is issued with a data port specified that is less than 1024 or greater than 65535.

4050 UDP Bomb (Attack, Atomic)
Triggers when the UDP length specified is less than the IP length specified.

4100 Tftp Passwd File (Attack, Compound)
Triggers on an attempt to access the passwd file (typically /etc/passwd) via TFTP.

6100 RPC Port Registration (Info, Atomic*)
Triggers when attempts are made to register new RPC services on a target host.

6101 RPC Port Unregistration (Info, Atomic*)
Triggers when attempts are made to unregister existing RPC services on a target host.

6102 RPC Dump (Info, Atomic*)
Triggers when an RPC dump request is issued to a target host.

6103 Proxied RPC Request (Attack, Atomic*)
Triggers when a proxied RPC request is sent to the portmapper of a target host.

6150 ypserv Portmap Request (Info, Atomic*)
Triggers when a request is made to the portmapper for the YP server daemon (ypserv) port.

6151 ypbind Portmap Request (Info, Atomic*)
Triggers when a request is made to the portmapper for the YP bind daemon (ypbind) port.

6152 yppasswdd Portmap Request (Info, Atomic*)
Triggers when a request is made to the portmapper for the YP password daemon (yppasswdd) port.

6153 ypupdated Portmap Request (Info, Atomic*)
Triggers when a request is made to the portmapper for the YP update daemon (ypupdated) port.

6154 ypxfrd Portmap Request (Info, Atomic*)
Triggers when a request is made to the portmapper for the YP transfer daemon (ypxfrd) port.

6155 mountd Portmap Request (Info, Atomic*)
Triggers when a request is made to the portmapper for the mount daemon (mountd) port.

6175 rexd Portmap Request (Info, Atomic*)
Triggers when a request is made to the portmapper for the remote execution daemon (rexd) port.

6180 rexd Attempt (Info, Atomic*)
Triggers when a call to the rexd program is made. The remote execution daemon is the server responsible for remote program execution. This may be indicative of an attempt to gain unauthorized access to system resources.

6190 statd Buffer Overflow (Attack, Atomic*)
Triggers when a large statd request is sent. This could be an attempt to overflow a buffer and gain access to system resources.

8000 FTP Retrieve Password File (Attack, Atomic*)
SubSig ID: 2101

Triggers on string "passwd" issued during an FTP session. May indicate someone attempting to retrieve the password file from a machine in order to crack it and gain unauthorized access to system resources.

Cisco Secure IS IDS Configuration Task List

See the following sections for configuration tasks for the Cisco Secure Integrated Software Intrusion Detection System feature. Each task in the list indicates if it is optional or required:

- Initializing Cisco Secure IS IDS (Required)
- Initializing the Post Office (Required)
- Configuring and Applying Audit Rules (Required)
- Verifying the Configuration (Optional)

For examples using the commands in this chapter, see the "Cisco Secure IS IDS Configuration Examples" section at the end of this chapter.

Initializing Cisco Secure IS IDS

To initialize Cisco Secure Integrated Software IDS on a router, use the following commands in global configuration mode:

Step	Command	Purpose
1	`router(conf)#ip audit smtp spam recipients`	Use the **ip audit smtp** command to set the threshold beyond which spamming in e-mail messages is suspected. Here, *recipients* is the maximum number of recipients in an e-mail message. The default is 250.
2	`router(conf)#ip audit po max-events number_events`	Use the **ip audit po max-events** command to set the threshold beyond which cued events are dropped from the cue for sending to the Cisco Secure IDS Director. Here, *number_events* is the number of events in the event cue. The default is 100. Increasing this number may have an impact on memory and performance, as each event in the event cue requires 32 KB of memory.
3	`router(conf)#exit`	Leaves global configuration mode.

Initializing the Post Office

Note You must reload the router every time you make a Post Office configuration change.

To initialize the Post Office system, use the following commands in global configuration mode:

Step	Command	Purpose

1	`router(conf)#ip audit notify nr-director`
	or
	`router(conf)#ip audit notify log`
		Use the **ip audit notify** command to send event notifications (alarms) to either a Cisco Secure IDS Director or a syslog server, or both.
		For example, if you are sending alarms to a Cisco Secure IDS Director, use the **nr-director** keyword in the command syntax. If you are sending alarms to a syslog server, use the **log** keyword in the command syntax.

2	`router(conf)#ip audit po local hostid host-id orgid org-id`	If you are sending alarms to a Cisco Secure IDS Director, you must set the Post Office parameters for both the router (using the **ip audit po local** command) and the Cisco Secure IDS Director (using the **ip audit po remote** command). Here, *host-id* is a unique number between 1 and 65535 that identifies the router, and *org-id* is a unique number between 1 and 65535 that identifies the organization to which the router and Director both belong.

Step	Command	Purpose
3	`router(conf)#ip audit po remote hostid host-id orgid org-id rmtaddress ip-address localaddress ip-address port port-number preference preference-number timeout seconds application application-type`	If you are sending alarms to a Cisco Secure IDS Director, you must also set the Post Office parameters for both the Cisco Secure IDS Director (using the **ip audit po remote** command). • *host-id* is a unique number between 1 and 65535 that identifies the Director. • *org-id* is a unique number between 1 and 65535 that identifies the organization to which the router and Director both belong. • **rmtaddress** *ip-address* is the Director's IP address. • **localaddress** *ip-address* is the router's interface IP address. • *port-number* identifies the UDP port on which the Director is listening for alarms (45000 is the default). • *preference-number* is the relative priority of the route to the Director (1 is the default)—if more than one route is used to reach the same Director, then one must be a primary route (preference 1) and the other a secondary route (preference 2). • *seconds* is the number of seconds the Post Office waits before it determines that a connection has timed out (5 is the default). • *application-type* is either **director** or **logger**. **Note** If you are sending Post Office notifications to a Sensor, use **logger** instead of **director** as your application. Sending to a logging application means that no alarms are sent to a GUI; instead, the Cisco Secure IDS alarm data is written to a flat file, which can then be processed with filters, such as **perl** and **awk**, or staged to a database. Use **logger** only in advanced applications where you want the alarms only to be logged and not displayed.
4	`router(conf)#logging console info`	If you are sending alarms to the syslog console, you have the option of seeing the syslog messages on the router console.
5	`router# exit`	Leaves global configuration mode.
6	`router# wr mem`	Saves the configuration.
7	`router# reload#`	Reloads the router.

After you have configured the router, add the Cisco Secure IS IDS router's Post Office information to the */usr/nr/etc/hosts* and */usr/nr/etc/routes* files on the Cisco Secure IDS Sensors and Directors communicating with the router. You can do this with the nrConfigure tool in Cisco Secure IDS. For more information, refer to the *NetRanger User Guide*.

Configuring and Applying Audit Rules

To configure and apply audit rules, use the following commands starting in global configuration mode:

Step	Command	Purpose

1	`router(conf)#ip audit info {action [alarm] [drop] [reset]}`
	`router(conf)#ip audit attack {action [alarm] [drop] [reset]}`
		Use the **ip audit info action** and **ip audit attack action** commands to set the default actions for info and attack signatures. Both types of signatures can take any or all of the following actions: alarm, drop, and reset. The default action is **alarm**.

2	`router(conf)#ip audit name audit-name {info | attack} [list standard-acl] [action [alarm] [drop] [reset]]`	Use the **ip audit name** command to create audit rules, where *audit-name* is a user-defined name for an audit rule. For example:
		`ip audit name audit-name info`
		`ip audit name audit-name attack`
		The default action is **alarm**.
		Note Use the same name when you assign attack and info type signatures.
		You can also use the **ip audit name** command to attach access control lists to an audit rule for filtering out sources of false alarms. In this case *standard-acl* is an integer representing an ACL. If you attach an ACL to an audit rule, the ACL must be defined as well:
		`ip audit name audit-name {info|attack} list acl-list`
		In the following example, ACL 99 is attached to the audit rule INFO, and ACL 99 is defined:
		`ip audit name INFO info list 99`
		`access-list 99 deny 10.1.1.0 0.0.0.255`
		`access-list 99 permit any`
		Note The ACL in the preceding example is *not* denying traffic from the 10.1.1.0 network (as expected if it were applied to an interface). Instead, the hosts on that network are not filtered through the audit process because they are trusted hosts. On the other hand, all other hosts, as defined by **permit any**, are processed by the audit rule.

Step	Command	Purpose

3	`router(conf)#ip audit signature signature-id {disable | list acl-list}`	Use the **ip audit signature** command to disable individual signatures. Disabled signatures are not included in audit rules, as this is a global configuration change:
		`ip audit signature signature-number disable`
		To re-enable a disabled signature, use the **no ip audit signature** command, where *signature-number* is the number of the disabled signature.
		You can also use the **ip audit signature** command to apply ACLs to individual signatures for filtering out sources of false alarms. In this case *signature-number* is the number of a signature, and *acl-list* is an integer representing an ACL:
		`ip audit signature signature-number list acl-list`
		For example, ACL 35 is attached to the 1234 signature, and then defined:
		`ip audit signature 1234 list 35`
		`access-list 35 deny 10.1.1.0 0.0.0.255`
		`access-list 35 permit any`
		Note The ACL in the preceding example is *not* denying traffic from the 10.1.1.0 network (as expected if it were applied to an interface). Instead, the hosts on that network are not filtered through the signature because they are trusted hosts or are otherwise causing false positives to occur. On the other hand, all other hosts, as defined by **permit any**, are processed by the signature.

4	`router(conf)# interface interface-number`	Enters interface configuration mode.
5	`router(conf)# ip audit audit-name {in | out}`	Use the **ip audit** command to apply an audit rule at an interface. With this command, *audit-name* is the name of an existing audit rule, and *direction* is either **in** or **out**.
6	`router(conf)# exit`	Leaves interface configuration mode.
7	`router(conf)# ip audit po protected ip-addr [to ip-addr]`	After you apply the audit rules to the router interfaces, use the **ip audit po protected** command to configure which network should be protected by the router. Here, *ip_addr* is the IP address to protect.
8	`router(conf)#exit`	Leaves global configuration mode.

Verifying the Configuration

You can verify that Cisco Secure IS IDS is properly configured with the **show ip audit configuration** command (see Example 1).

Example 1 Output from show ip audit configuration Command

```
ids2611# show ip audit configuration
Event notification through syslog is enabled
Event notification through Net Director is enabled
Default action(s) for info signatures is alarm
Default action(s) for attack signatures is alarm drop reset
Default threshold of recipients for spam signature is 25
PostOffice:HostID:55 OrgID:123 Msg dropped:0
          :Curr Event Buf Size:100  Configured:100
HID:14 OID:123 S:1 A:2 H:82 HA:49 DA:0 R:0 Q:0
 ID:1 Dest:10.1.1.99:45000 Loc:172.16.58.99:45000 T:5 S:ESTAB *

Audit Rule Configuration
 Audit name AUDIT.1
    info actions alarm
    attack actions alarm drop reset
```

You can verify which interfaces have audit rules applied to them with the **show ip audit interface** command (see Example 2).

Example 2 Output from show ip audit interface Command

```
ids2611# show ip audit interface
Interface Configuration
 Interface Ethernet0
  Inbound IDS audit rule is AUDIT.1
    info actions alarm
    attack actions alarm drop reset
  Outgoing IDS audit rule is not set
 Interface Ethernet1
  Inbound IDS audit rule is AUDIT.1
    info actions alarm
    attack actions alarm drop reset
  Outgoing IDS audit rule is not set
```

Monitoring and Maintaining Cisco Secure IS IDS

This section describes the EXEC commands used to monitor and maintain Cisco Secure Integrated Software IDS.

Command	Purpose
`router# clear ip audit configuration`	Disables Cisco Secure IS IDS, removes all intrusion detection configuration entries, and releases dynamic resources.
`router# clear ip audit statistics`	Resets statistics on packets analyzed and alarms sent.
`router# show ip audit statistics`	Displays the number of packets audited and the number of alarms sent, among other information.

The following display provides sample output from the **show ip audit statistics** command:

```
Signature audit statistics [process switch:fast switch]
  signature 2000 packets audited: [0:2]
  signature 2001 packets audited: [9:9]
  signature 2004 packets audited: [0:2]
  signature 3151 packets audited: [0:12]
Interfaces configured for audit 2
Session creations since subsystem startup or last reset 11
Current session counts (estab/half-open/terminating) [0:0:0]
Maxever session counts (estab/half-open/terminating) [2:1:0]
Last session created 19:18:27
Last statistic reset never

HID:1000 OID:100 S:218 A:3 H:14085 HA:7114 DA:0 R:0
```

Cisco Secure IS IDS Configuration Examples

The following sections provide Cisco Secure IS IDS configuration examples:

- Cisco Secure IS IDS Reporting to Two Directors Example
- Adding an ACL to the Audit Rule Example
- Disabling a Signature Example
- Adding an ACL to Signatures Example
- Dual-Tier Signature Response Example

Cisco Secure IS IDS Reporting to Two Directors Example

In the following example, Cisco Secure Integrated Software IDS is initialized. Notice that the router is reporting to two Directors. Also notice that the AUDIT.1 audit rule will apply both info and attack signatures:

```
ip audit smtp spam 25
ip audit notify nr-director
ip audit notify log
ip audit po local hostid 55 orgid 123
ip audit po remote hostid 14 orgid 123 rmtaddress 10.1.1.99 localaddress 10.1.1.1
ip audit po remote hostid 15 orgid 123 rmtaddress 10.1.2.99 localaddress 10.1.1.1

ip audit name AUDIT.1 info action alarm
ip audit name AUDIT.1 attack action alarm drop reset

interface e0
   ip address 10.1.1.1 255.0.0.0
   ip audit AUDIT.1 in

interface e1
   ip address 172.16.57.1 255.255.255.0
   ip audit AUDIT.1 in
```

Adding an ACL to the Audit Rule Example

In the following example, an ACL is added to account for a Cisco Secure IDS Scanner (172.16.59.16) that scans for all types of attacks. As a result, no packets originating from the device will be audited:

```
ip audit smtp spam 25
ip audit notify nr-director
ip audit notify log
ip audit po local hostid 55 orgid 123
ip audit po remote hostid 14 orgid 123 rmtaddress 10.1.1.99 localaddress 10.1.1.1
ip audit po remote hostid 15 orgid 123 rmtaddress 10.1.2.99 localaddress 10.1.1.1

ip audit name AUDIT.1 info list 90 action alarm
ip audit name AUDIT.1 attack list 90 action alarm drop reset

interface e0
   ip address 10.1.1.1 255.0.0.0
   ip audit AUDIT.1 in

interface e1
   ip address 172.16.57.1 255.255.255.0
   ip audit AUDIT.1 in

access-list 90 deny 172.16.59.16
access-list 90 permit any
```

Disabling a Signature Example

The security administrator notices that the router is generating a lot of false positives for signatures 1234, 2345, and 3456. The system administrator knows that there is an application on the network that is causing signature 1234 to fire, and it is not an application that should cause security concerns. This signature can be disabled, as illustrated in the following example:

```
ip audit smtp spam 25
ip audit notify nr-director
ip audit notify log
ip audit po local hostid 55 orgid 123
ip audit po remote hostid 14 orgid 123 rmtaddress 10.1.1.99 localaddress 10.1.1.1
ip audit po remote hostid 15 orgid 123 rmtaddress 10.1.2.99 localaddress 10.1.1.1


ip audit signature 1234 disable

ip audit name AUDIT.1 info list 90 action alarm
ip audit name AUDIT.1 attack list 90 action alarm drop reset

interface e0
   ip address 10.1.1.1 255.0.0.0
   ip audit AUDIT.1 in

interface e1
   ip address 172.16.57.1 255.255.255.0
   ip audit AUDIT.1 in

access-list 90 deny 172.16.59.16
access-list 90 permit any
```

Adding an ACL to Signatures Example

After further investigation, the security administrator discovers that the false positives for signatures 2345 and 3456 are caused by specific applications on hosts 10.4.1.1 and 10.4.1.2, as well as by some workstations using DHCP on the 172.16.58.0 subnetwork. Attaching an ACL that denies processing of these hosts stops the creation of false positive alarms, as illustrated in the following example:

```
ip audit smtp spam 25
ip audit notify nr-director
ip audit notify log
ip audit po local hostid 55 orgid 123
ip audit po remote hostid 14 orgid 123 rmtaddress 10.1.1.99 localaddress 10.1.1.1
ip audit po remote hostid 15 orgid 123 rmtaddress 10.1.2.99 localaddress 10.1.1.1

ip audit signature 1234 disable
ip audit signature 2345 list 91
ip audit signature 3456 list 91

ip audit name AUDIT.1 info list 90 action alarm
ip audit name AUDIT.1 attack list 90 action alarm drop reset

interface e0
   ip address 10.1.1.1 255.0.0.0
   ip audit AUDIT.1 in

interface e1
   ip address 172.16.57.1 255.255.255.0
   ip audit AUDIT.1 in

access-list 90 deny 172.16.59.16
access-list 90 permit any
access-list 91 deny host 10.4.1.1
access-list 91 deny host 10.4.1.2
access-list 91 deny 172.16.58.0 0.0.0.255
access-list 91 permit any
```

Dual-Tier Signature Response Example

The company has now reorganized and has placed only trusted people on the 172.16.57.0 network. The work done by the employees on these networks must not be disrupted by Cisco Secure Integrated Software IDS, so attack signatures in the AUDIT.1 audit rule now will only alarm on a match.

For sessions that originate from the outside network, any attack signature matches (other than the false positive ones that are being filtered out) are to be dealt with in the following manner: send an alarm, drop the packet, and reset the TCP session.

This dual-tier method of signature response is accomplished by configuring two different audit specifications and applying each to a different ethernet interface, as illustrated in the following example:

```
ip audit smtp spam 25
ip audit notify nr-director
ip audit notify log
ip audit po local hostid 55 orgid 123
ip audit po remote hostid 14 orgid 123 rmtaddress 10.1.1.99 localaddress 10.1.1.1
ip audit po remote hostid 15 orgid 123 rmtaddress 10.1.2.99 localaddress 10.1.1.1

ip audit signature 1234 disable
ip audit signature 2345 list 91
ip audit signature 3456 list 91

ip audit name AUDIT.1 info list 90 action alarm
ip audit name AUDIT.1 attack list 90 action alarm
ip audit name AUDIT.2 info action alarm
ip audit name AUDIT.2 attack alarm drop reset

interface e0
   ip address 10.1.1.1 255.0.0.0
   ip audit AUDIT.2 in

interface e1
   ip address 172.16.57.1 255.255.255.0
   ip audit AUDIT.1 in

access-list 90 deny host 172.16.59.16
access-list 90 permit any
access-list 91 deny host 10.4.1.1
access-list 91 deny host 10.4.1.2
access-list 91 deny 172.16.58.0 0.0.0.255
access-list 91 permit any
```

Configuring Authentication Proxy

This chapter describes the Cisco Secure Integrated Software Authentication Proxy feature. Authentication proxy provides dynamic, per-user authentication and authorization, authenticating users against industry standard TACACS+ and RADIUS authentication protocols. Authenticating and authorizing connections by user provides more robust protection against network attacks.

For a complete description of the authentication proxy commands in this chapter, refer to the "Authentication Proxy Commands" chapter of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

In This Chapter

This chapter has the following sections:

- Authentication Proxy Overview
- Authentication Proxy Configuration Task List
- Monitoring and Maintaining the Authentication Proxy
- Authentication Proxy Configuration Examples

Authentication Proxy Overview

The Cisco Secure Integrated Software authentication proxy feature allows network administrators to apply specific security policies on a per-user basis. Previously, user identity and related authorized access was associated with a user's IP address, or a single security policy had to be applied to an entire user group or sub network. Now, users can be identified and authorized on the basis of their per-user policy, and access privileges tailored on an individual basis are possible, as opposed to general policy applied across multiple users.

With the authentication proxy feature, users can log in to the network or access the Internet via HTTP, and their specific access profiles are automatically retrieved and applied from a CiscoSecure ACS, or other RADIUS, or TACACS+ authentication server. The user profiles are active only when there is active traffic from the authenticated users.

The authentication proxy is compatible with other Cisco IOS security features such as Network Address Translation (NAT), Context-based Access Control (CBAC), IP Security (IPSec) encryption, and Cisco Secure VPN Client (VPN client) software.

This section has the following sections:

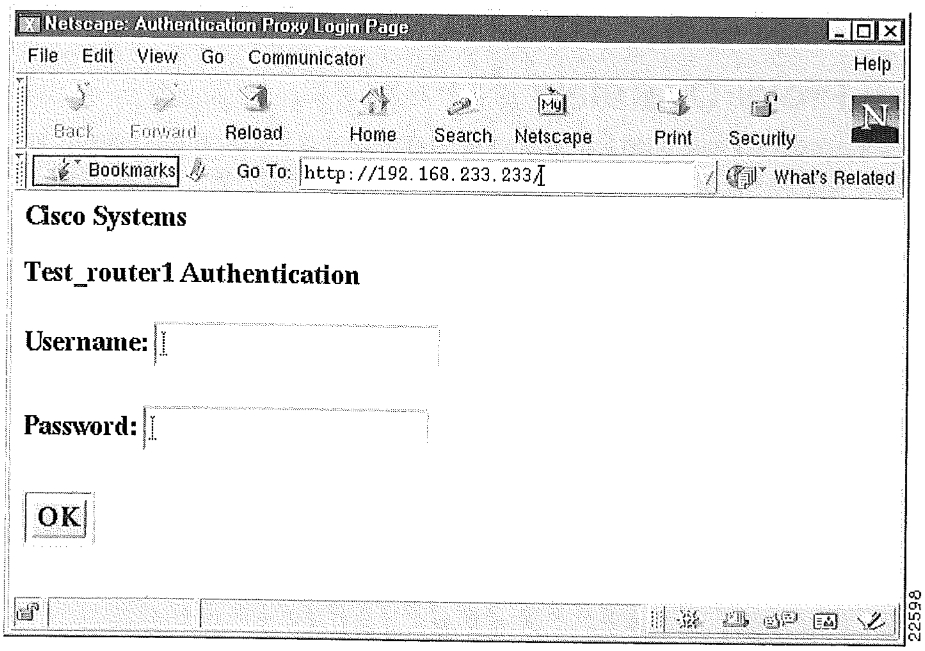
- How the Authentication Proxy Works
- Secure Authentication
- Using the Authentication Proxy
- When to Use the Authentication Proxy
- Applying the Authentication Proxy
- Operation with One-Time Passwords
- Compatibility with Other Security Features
- Protection Against Denial-of-Service Attacks
- Risk of Spoofing with Authentication Proxy
- Comparison with the Lock-and-Key Feature
- Restrictions
- Prerequisites to Configuring Authentication Proxy

How the Authentication Proxy Works

When a user initiates an HTTP session through the firewall, the authentication proxy is triggered. The authentication proxy first checks to see if the user has been authenticated. If a valid authentication entry exists for the user, the connection is completed with no further intervention by the authentication proxy. If no entry exists, the authentication proxy responds to the HTTP connection request by prompting the user for a username and password.

Figure 18 illustrates the authentication proxy HTML login page.

Figure 18 Authentication Proxy Login Page



http://192.168.233.233/

Cisco Systems

Test_router1 Authentication

Username:

Password:

Users must successfully authenticate with the authentication server by entering a valid username and password.

If the authentication succeeds, the user's authorization profile is retrieved from the AAA server. The authentication proxy uses the information in this profile to create dynamic access control entries (ACEs) and add them to the inbound (input) access control list (ACL) of an input interface, and to the outbound (output) ACL of an output interface, if an output ACL exists at the interface. By doing this, the firewall allows authenticated users access to the network as permitted by the authorization profile. For example, a user can initiate a Telnet connection through the firewall if Telnet is permitted in the user's profile.

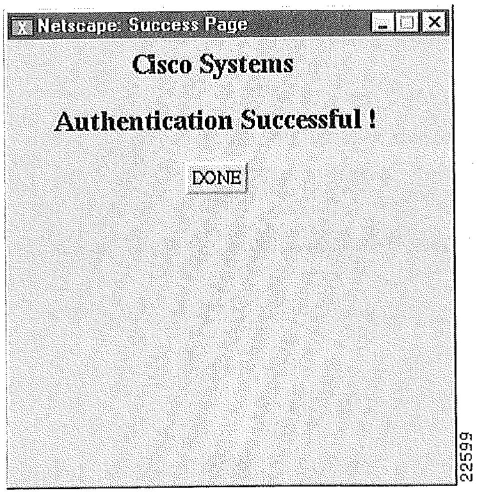
If the authentication fails, the authentication proxy reports the failure to the user, and prompts the user with multiple retries. If the user fails to authenticate after five attempts, the user must wait two minutes and initiate another HTTP session to trigger authentication proxy.

The login page is refreshed each time the user makes requests to access information from a web server.

The authentication proxy customizes each of the access list entries in the user profile by replacing the source IP addresses in the downloaded access list with the source IP address of the authenticated host.

At the same time that dynamic ACEs are added to the interface configuration, the authentication proxy sends a message to the user confirming that the login was successful. Figure 19 illustrates the login status in the HTML page.

Figure 19 Authentication Proxy Login Status Message



Cisco Systems

Authentication Successful !

The authentication proxy sets up an inactivity (idle) timer for each user profile. As long as there is activity through the firewall, new traffic initiated from the user's host does not trigger the authentication proxy, and authorized user traffic is permitted access through the firewall.

If the idle timer expires, the authentication proxy removes the user's profile information and dynamic access lists entries. When this happens, traffic from the client host is blocked. The user must initiate another HTTP connection to trigger the authentication proxy.

Secure Authentication

The authentication proxy uses JavaScript to help achieve secure authentication using the client browser. Secure authentication prevents a client from mistakenly submitting a username and password to a network web server other than the authentication proxy router.

This section has the following sections:

- Operation with JavaScript
- Operation without JavaScript

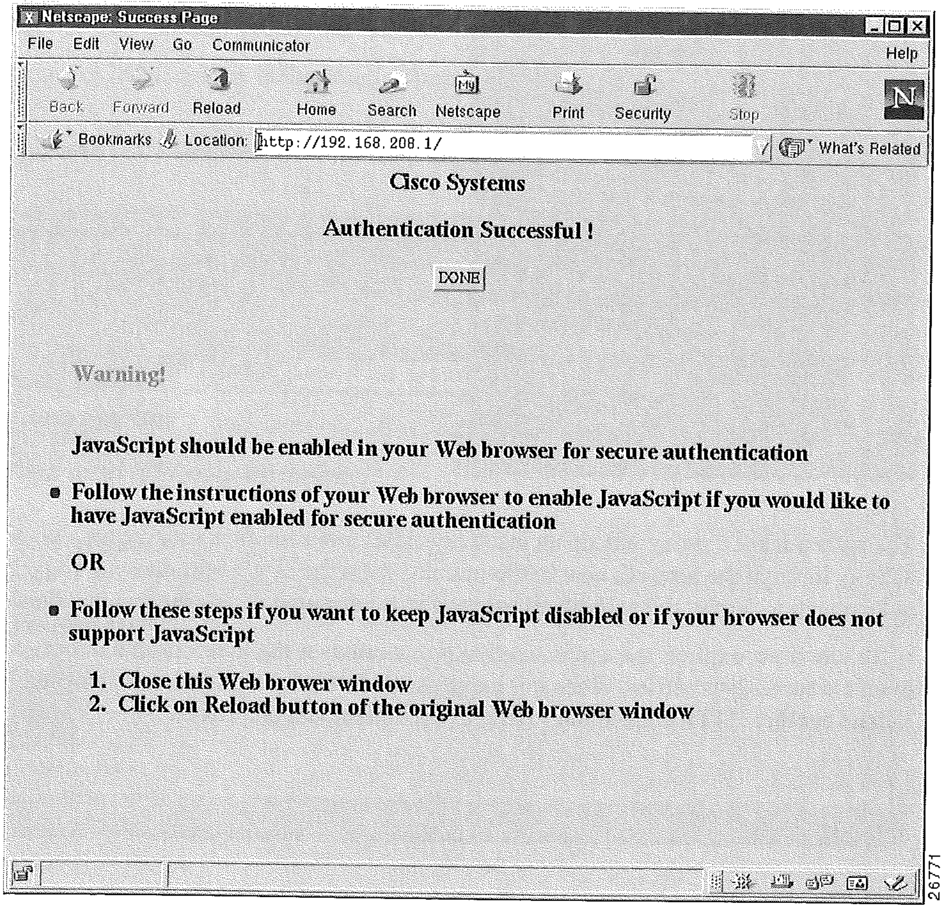
Operation with JavaScript

Users should enable JavaScript on the browser prior to initiating an HTTP connection. With JavaScript enabled on the browser, secure authentication is done automatically, and the user sees the authentication message shown in Figure 19. The HTTP connection is completed automatically for the user.

Operation without JavaScript

If the client browser does not support JavaScript, or if site security policy prevents users from enabling JavaScript, any login attempt generates a popup window with instructions for manually completing the connection. Figure 20 illustrates the authentication proxy login status message with JavaScript disabled on the browser.

Figure 20 Authentication Proxy Login Status Message with JavaScript Disabled


Netscape: Success Page
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://192.168.208.1/ What's Related
DONE
26771

Cisco Systems

Authentication Successful !

Warning!

JavaScript should be enabled in your Web browser for secure authentication

- Follow the instructions of your Web browser to enable JavaScript if you would like to have JavaScript enabled for secure authentication

OR

- Follow these steps if you want to keep JavaScript disabled or if your browser does not support JavaScript
  1. Close this Web brower window
  2. Click on Reload button of the original Web browser window

To close this window, click Close on the browser File menu.

After closing the popup window, the user should click Reload (Refresh for Internet Explorer) on the browser window in which the authentication login page is displayed. If the user's last authentication attempt succeeds, clicking Reload brings up the web page the user is trying to retrieve. If the user's last attempt fails, clicking Reload causes the authentication proxy to intercept the client HTTP traffic again, prompting the user with another login page soliciting the username and password.

If JavaScript is not enabled, it is strongly recommended that site administrators advise users of the correct procedure for closing the popup window as described in the "Establishing User Connections with JavaScript Disabled" section.

Using the Authentication Proxy

Unlike some Cisco Secure Integrated Software features that operate transparently to the user, the authentication proxy feature requires some user interaction on the client host. Table 20 describes the interaction of the authentication proxy with the client host.

Table 20 Authentication Proxy Interaction with the Client Host

Authentication Proxy Action with Client	Description
Triggering on HTTP connections	If a user is not currently authenticated at the firewall router, any HTTP connection initiated by the user triggers the authentication proxy. If the user is already authenticated, the authentication proxy is transparent to the user.
Logging in using the login page	Triggering the authentication proxy generates an HTML-based login page.The user must enter a username and password to authenticate with the AAA server. Figure 18 illustrates the authentication proxy login page.
Authenticating the user at the client	Following the login attempt, the authentication proxy action can vary depending on whether JavaScript is enabled in the browser. If JavaScript is enabled, and authentication is successful, the authentication proxy displays a message indicating the status of the authentication as shown in Figure 19. After displaying the authentication status, the proxy automatically completes the HTTP connection. If JavaScript is disabled, and authentication is successful, the authentication proxy generates a popup window with additional instructions for completing the connection. See Figure 20. If authentication is unsuccessful in any case, the user must log in again from the login page.

When to Use the Authentication Proxy

Here are a few examples of when you might use the authentication proxy:

- You want to manage access privileges on an individual (per-user) basis using the services provided by the authentication servers instead of configuring access control based on host IP address or global access policies. Authenticating and authorizing users from any host IP address also allows network administrators to configure host IP addresses using DHCP.
- You want to authenticate and authorize local users before permitting access to intranet or Internet services or hosts through the firewall.
- You want to authenticate and authorize remote users before permitting access to local services or hosts through the firewall.
- You want to control access for specific extranet users. For example, you might want to authenticate and authorize the financial officer of a corporate partner with one set of access privileges, while authorizing the technology officer for that same partner to use another set of access privileges.
- You want to use the authentication proxy in conjunction with VPN client software to validate users and to assign specific access privileges.

Applying the Authentication Proxy

Apply the authentication proxy in the inbound direction at any interface on the router where you want per-user authentication and authorization. Applying the authentication proxy inbound at an interface causes it to intercept a user's initial connection request before that request is subjected to any other processing by the firewall. If the user fails to authenticate with the AAA server, the connection request is dropped.

How you apply the authentication proxy depends on your security policy. For example, you can block all traffic through an interface, and enable the authentication proxy feature to require authentication and authorization for all user initiated HTTP connections. Users are authorized for services only after successful authentication with the AAA server.

The authentication proxy feature also allows you to use standard access lists to specify a host or group of hosts whose initial HTTP traffic triggers the proxy.

Figure 21 shows the authentication proxy applied at the LAN interface with all network users required to authenticate upon the initial connection (all traffic is blocked at each interface).

Figure 21 **Applying the Authentication Proxy at the Local Interface**

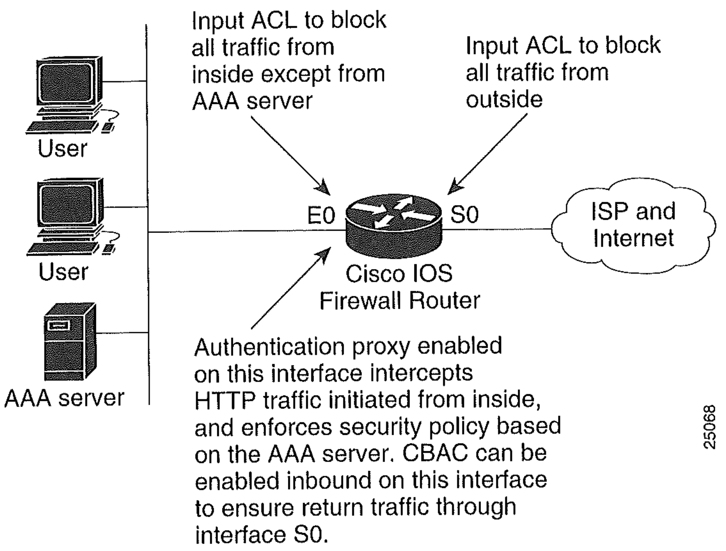
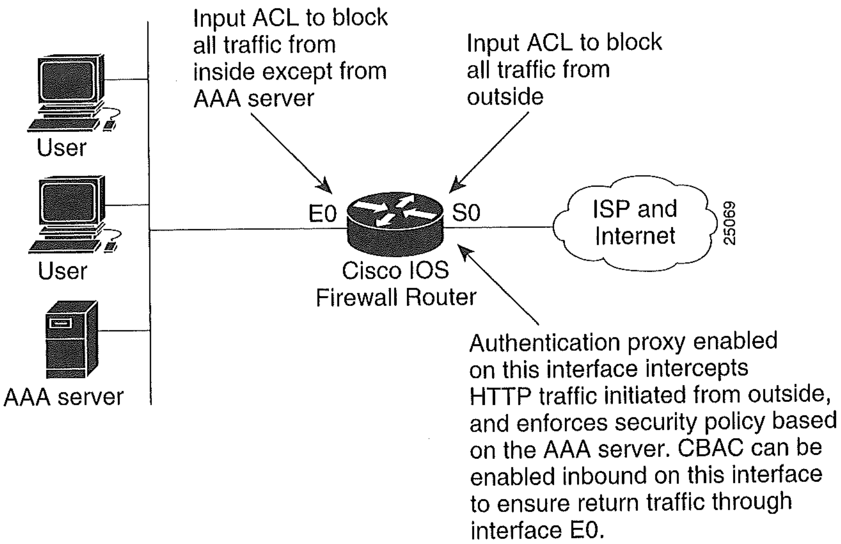

Input ACL to block all traffic from inside except from AAA server
Input ACL to block all traffic from outside
User
User
AAA server
E0
S0
ISP and Internet
Cisco IOS Firewall Router
Authentication proxy enabled on this interface intercepts HTTP traffic initiated from inside, and enforces security policy based on the AAA server. CBAC can be enabled inbound on this interface to ensure return traffic through interface S0.
25068

Figure 22 shows the authentication proxy applied at the dial-in interface with all network traffic blocked at each interface.

Figure 22 **Applying the Authentication Proxy at an Outside Interface**


Input ACL to block all traffic from inside except from AAA server
Input ACL to block all traffic from outside
User
User
AAA server
E0
S0
ISP and Internet
25069
Cisco IOS Firewall Router
Authentication proxy enabled on this interface intercepts HTTP traffic initiated from outside, and enforces security policy based on the AAA server. CBAC can be enabled inbound on this interface to ensure return traffic through interface E0.

Operation with One-Time Passwords

Using a one-time password, users enter the username and one-time password in the HTML login page as usual.

Users must enter the correct token password within the first three attempts. After three incorrect entries, the user must enter two valid token passwords in succession before authentication is granted with the AAA server.

Compatibility with Other Security Features

The authentication proxy is compatible with Cisco IOS software and with Cisco IOS security features:

- Cisco Secure IS Intrusion Detection System (IDS)
- NAT
- CBAC
- IPSec encryption
- VPN client software

The authentication proxy works transparently with the Cisco Secure IS IDS and IPSec encryption features. The following sections describe the relationship of the NAT, CBAC, and VPN client software features with the authentication proxy.

- NAT Compatibility
- CBAC Compatibility
- VPN Client Compatibility

NAT Compatibility

The authentication proxy feature is compatible with NAT; however, to run successfully with NAT, you must configure CBAC.

For example, when dynamic NAT is configured, the client's IP address might be translated to different addresses during the time the user is authenticated with the authentication proxy. Assume that the client is running a HTTP session. The user's original IP address is 10.1.1.1, which is translated by NAT to 192.168.2.2. NAT guarantees that during this session, 10.1.1.1 is always translated to 192.168.2.2. When the user is first authenticated, a set of dynamic ACEs is created to support the user. Subsequent sessions can use different NAT addresses, which are not covered by the original dynamic ACEs created by the authentication proxy. In this case, it is strongly recommended that you configure CBAC to take care of the translated addresses and to create the matching ACEs.

CBAC ensures that the translated address for the session is associated with the original host address.

CBAC Compatibility

To run successfully in all configurations, and to ensure return traffic for authorized user connections is permitted through the firewall, configure CBAC with the authentication proxy.

Because the authentication proxy does not create ACEs to support return traffic or data channel traffic, you must either create static ACLs to allow the return traffic or configure CBAC inspection rules in the firewall configuration.

VPN Client Compatibility

Using the authentication proxy, network administrators can apply an extra layer of security and access control for VPN client traffic. If a VPN client initiates an HTTP connection, the authentication proxy first checks for prior client authentication. If the client is authenticated, authorized traffic is permitted. If the client is not authenticated, the HTTP request triggers the authentication proxy, and the user is prompted for a username and password.

If the user authentication is successful, the authentication proxy retrieves the user profile from the AAA server. The source address in the user profile entries is replaced with the IP address of the authenticated VPN client from the decrypted packet.

Protection Against Denial-of-Service Attacks

The authentication proxy monitors the level of incoming HTTP requests. For each request, the authentication proxy prompts for the user's login credentials. A high number of open requests could indicate that the router is the subject of a denial-of-service (DoS) attack. The authentication proxy limits the level of open requests and drops additional requests until the number of open requests has dropped below 40.

If the firewall is experiencing a high level of connection requests requiring authentication, legitimate network users might experience delays when making connections, or the connection might be rejected and the user must try the connection again.

Risk of Spoofing with Authentication Proxy

When the authentication proxy is triggered, it creates a dynamic opening in the firewall by temporarily reconfiguring an interface with user access privileges. While this opening exists, another host might spoof the authenticated user's address to gain access behind the firewall. The authentication proxy does not cause the address spoofing problem; the problem is only identified here as a concern to the user. Spoofing is a problem inherent to all access lists, and the authentication proxy does not specifically address this problem.

Comparison with the Lock-and-Key Feature

Lock-and-key is another Cisco Secure Integrated Software feature that uses authentication and dynamic access list to provide user access through the firewall. Table 21 compares the authentication proxy and Lock-and-key features.

Table 21 Comparison of the Authentication Proxy and Lock-and-Key Features

Lock-and-Key	Authentication Proxy
Triggers on Telnet connection requests.	Triggers on HTTP connection requests.
TACACS+, RADIUS, or local authentication.	TACACS+ or RADIUS authentication and authorization.
Access lists are configured on the router only.	Access lists are retrieved from the AAA server only.
Access privileges are granted based on the user's host IP address.	Access privileges are granted on a per-user and host IP address basis.
Access lists are limited to one entry for each host IP address.	Access lists can have multiple entries as defined by the user profiles on the AAA server.
Associates a fixed IP addresses with a specific user. Users must log in from the host with that IP address.	Allows DHCP-based host IP addresses, meaning that users can log in from any host location and obtain authentication and authorization.

Use the authentication proxy in any network environment that provides a per-user security policy. Use Lock-and-key in network environments that might benefit from local authentication and a limited number of router-based access control policies based on host addresses. Use Lock-and-key in environments not using the Cisco Secure Integrated Software.

Restrictions

- The authentication proxy triggers only on HTTP connections.
- HTTP services must be running on the standard (well-known) port, which is port 80 for HTTP.
- The authentication proxy does not support AAA accounting.
- Client browsers must enable JavaScript for secure authentication.
- The authentication proxy access lists apply to traffic passing through the router. Traffic destined to the router is authenticated by the existing authentication methods provided by Cisco IOS software.
- The authentication proxy does not support concurrent usage; that is, if two users try to log in from the same host at the same time, authentication and authorization applies only to the user who first submits a valid username and password.
- Load balancing using multiple or different AAA servers is not supported.

Prerequisites to Configuring Authentication Proxy

Prior to configuring authentication proxy, review the following:

- For the authentication proxy to work properly, the client host must be running the following browser software:
  - Microsoft Internet Explorer 3.0 or later
  - Netscape Navigator 3.0 or later
- The authentication proxy has an option to use standard access lists. You must have a solid understanding of how access lists are used to filter traffic before you attempt to configure the authentication proxy. For an overview of how to use access lists with the Cisco Secure Integrated Software, refer to the chapter "Access Control Lists: Overview and Guidelines."
- The authentication proxy employs user authentication and authorization as implemented in Cisco's authentication, authorization, and accounting (AAA) paradigm. You must understand how to configure AAA user authentication and authorization before you configure the authentication proxy. User authentication and authorization is explained in the chapter, "Authentication, Authorization, and Accounting (AAA)."
- To run the authentication proxy successfully with Cisco Secure Integrated Software, configure CBAC on the firewall. For complete information on the CBAC feature, refer to "Cisco Secure Integrated Software Feature Set" in the Cisco IOS Release 12.0 New Features section on Cisco Connection Online (CCO).

Authentication Proxy Configuration Task List

To configure the authentication proxy feature, perform the following tasks:

- Configuring AAA (Required)
- Configuring the HTTP Server (Required)
- Configuring the Authentication Proxy (Required)
- Verifying the Authentication Proxy (Optional)

For authentication proxy configuration examples using the commands in this chapter, refer to the "Authentication Proxy Configuration Examples" section at the end of this chapter.

Configuring AAA

You must configure the authentication proxy for AAA services. Use the following commands in global configuration mode to enable authorization and to define the authorization methods:

Step	Command	Purpose
1	router(config)# **aaa new-model**	Enables the AAA functionality on the router.
2	router(config)# **aaa authentication login default** *TACACS+ RADIUS*	Defines the list of authentication methods at login.
3	router(config)# **aaa authorization auth-proxy default** [*method1* [*method2*...]]	Use the **auth-proxy** keyword to enable authentication proxy for AAA methods.
4	router(config)# **tacacs-server host** *hostname*	Specifies an AAA server. For RADIUS servers, use the **radius server host** command.
5	router(config)# **tacacs-server key** *sting*	Sets the authentication and encryption key for communications between the router and the AAA server. For RADIUS servers use the **radius server key** command.
6	router(config)# **access-list** *access-list-number* **permit tcp host** *source* **eq** *tacacs* **host** *destination*	Creates an ACL entry to allow the AAA server return traffic to the firewall. The source address is the IP address of the AAA server, and the destination is the IP address of the router interface where the AAA server resides.

In addition to configuring AAA on the firewall router, the authentication proxy requires a per-user access profile configuration on the AAA server. To support the authentication proxy, configure the AAA authorization service "auth-proxy" on the AAA server as outlined here:

- Define a separate section of authorization for **auth-proxy** to specify the downloadable user profiles. This does not interfere with other type of services, such as EXEC. The following example shows a user profile on a TACACS server:

```
default authorization = permit
key = cisco
user = newuser1 {
login = cleartext cisco
service = auth-proxy
{
priv-lvl=15
proxyacl#1="permit tcp any any eq 26"
proxyacl#2="permit icmp any host 60.0.0.2"
proxyacl#3="permit tcp any any eq ftp"
proxyacl#4="permit tcp any any eq ftp-data"
proxyacl#5="permit tcp any any eq smtp"
proxyacl#6="permit tcp any any eq telnet"
}
}
```

- The only supported attribute in the AAA server user configuration is **proxyacl#*n***. Use the **proxyacl#*n*** attribute when configuring the access lists in the profile. The attribute **proxyacl#*n*** is for both RADIUS and TACACS+ attribute-value (AV) pairs.
- The privilege level must be set to 15 for all users.
- The access lists in the user profile on the AAA server must have **permit** only access commands.
- Set the source address to **any** in each of the user profile access list entries. The source address in the access lists is replaced with the source address of the host making the authentication proxy request when the user profile is downloaded to the firewall.

- The supported AAA servers are:
  - — CiscoSecure ACS 2.1.x for Windows NT
  - — CiscoSecure ACS 2.3 for Windows NT
  - — CiscoSecure ACS 2.2.4 for UNIX
  - — CiscoSecure ACS 2.3 for UNIX
  - — TACACS+ server (vF4.02.alpha)
  - — Ascend RADIUS server - radius-980618 (required avpair patch)
  - — Livingston RADIUS server (v1.16)

Refer to the "AAA Server User Profile Example" section for sample AAA server configurations.

Configuring the HTTP Server

To use the authentication proxy, you must also enable the HTTP server on the firewall and set the HTTP server authentication method to use AAA. Enter the following commands in global configuration mode:

Step	Command	Purpose
1	`router(config)#` **`ip http server`**	Enables the HTTP server on the router. The authentication proxy uses the HTTP server to communicate with the client for user authentication.
2	`router(config)#` **`ip http authentication aaa`**	Sets the HTTP server authentication method to AAA.
3	`router(config)#` **`ip http access-class`** *`access-list-number`*	Specifies the access list for the HTTP server. Use the standard access list number configured in the "Interface Configuration" section.

Configuring the Authentication Proxy

Note Set the **auth-cache-time** option for any authentication proxy rule to a higher value than the idle timeout value for any CBAC inspection rule. When the authentication proxy removes an authentication cache along with its associated dynamic user ACL, there might be some idle connections monitored by CBAC, and removal of user-specific ACLs could cause those idle connections to hang. If CBAC has a shorter idle timeout, CBAC resets these connections when the idle timeout expires; that is, before the authentication proxy removes the user profile.

To configure the authentication proxy, use the following commands, beginning in global configuration mode:

Step	Command	Purpose
1	`router(config)# ip auth-proxy auth-cache-time min`	Sets the global authentication proxy idle timeout value in minutes. If the timeout expires, user authentication entries are removed, along with any associated dynamic access lists. The default value is 60 minutes.
2	`router(config)# ip auth-proxy auth-proxy-banner`	(Optional) Displays the name of the firewall router in the authentication proxy login page. The banner is disabled by default.
3	`router(config)# ip auth-proxy name auth-proxy-name http [auth-cache-time min] [list std-access-list`	Creates authentication proxy rules. The rules define how you apply authentication proxy. This command associates connections initiating HTTP protocol traffic with an authentication proxy name. You can associate the named rule with an access control list, providing control over which hosts use the authentication proxy feature. If no standard access list is defined, the named authentication proxy rule intercepts HTTP traffic from all hosts whose connection initiating packets are received at the configured interface. (Optional) The **auth-cache-time** option overrides the global authentication proxy cache timer. This option provides more control over timeout values for a specific authentication proxy rule. If no value is specified, the proxy rule assumes the value set with the **ip auth-proxy auth-cache-time** command. (Optional) The **list** option allows you to apply a standard access list to a named authentication proxy rule. HTTP connections initiated from hosts in the access list are intercepted by the authentication proxy.
4	`router(config)# interface type`	Enters interface configuration mode by specifying the interface type on which to apply the authentication proxy.
5	`router(config-if)# ip auth-proxy auth-proxy-name`	In interface configuration mode, applies the named authentication proxy rule at the interface. This command enables the authentication proxy rule with that name.

Verifying the Authentication Proxy

Verifying the authentication proxy configuration can have several components:

- Checking the Authentication Proxy Configuration (Optional)
- Establishing User Connections with JavaScript Enabled (Optional)
- Establishing User Connections with JavaScript Disabled (Optional)

Checking the Authentication Proxy Configuration

To check the current authentication proxy configuration, use the **show ip auth-proxy configuration** command in privileged EXEC mode:

Command	Purpose
`router# show ip auth-proxy configuration`	Displays the authentication proxy configuration.

In the following example, the global authentication proxy idle timeout value is set to 60 minutes, the named authentication proxy rule is "pxy," and the idle timeout value for this named rule is 1 minute. The display shows that no host list is specified, meaning that all connections initiating HTTP traffic at the interface are subject to the authentication proxy rule:

```
router# sh ip auth-proxy configuration
Authentication cache time is 60 minutes
Authentication Proxy Rule Configuration
Auth-proxy name pxy
http list not specified auth-cache-time 1 minutes
```

To verify that the authentication proxy is successfully configured on the router, ask a user to initiate an HTTP connection through the router. The user must have authentication and authorization configured at the AAA server. If the user authentication is successful, the firewall completes the HTTP connection for the user. If the authentication is unsuccessful, check the access list and the AAA server configurations.

Display the user authentication entries using the **show ip auth-proxy cache** command in privileged EXEC mode:

Command	Purpose
`router# show ip auth-proxy cache`	Displays the list of user authentication entries.

The authentication proxy cache lists the host IP address, the source port number, the timeout value for the authentication proxy, and the state of the connection. If the authentication proxy state is HTTP_ESTAB, the user authentication was successful.

```
router# show ip auth-proxy cache
Authentication Proxy Cache
 Client IP 192.168.25.215 Port 57882, timeout 1, state HTTP_ESTAB
```

Wait for one minute, which is the timeout value for this named rule, and ask the user to try the connection again. After one minute, the user connection is denied because the authentication proxy has removed the user's authentication entry and any associated dynamic ACLs. The user is presented with a new authentication login page and must log in again to gain access through the firewall.

Establishing User Connections with JavaScript Enabled

To verify client connections using the authentication proxy with JavaScript enabled on the client browser, follow this procedure:

Step 1 From a client host, initiate an HTTP connection through the firewall. This generates the authentication proxy login page.

Step 2 At the authentication proxy login page, enter a username and password.

Step 3 Click **OK** to submit the username and password to the AAA server.

A popup window appears indicating whether the login attempt succeeded or failed. If the authentication is successful, the connection is completed automatically. If the authentication fails, the authentication proxy reports the failure to the user, and prompts the user with multiple retries.

Note If the authentication attempt is unsuccessful after five attempts, the user must wait two minutes and initiate another HTTP session to trigger authentication proxy.

Establishing User Connections with JavaScript Disabled

The authentication proxy design requires JavaScript to ensure secure authentication. You can use the authentication proxy without enabling JavaScript on the browser, but this poses a potential security risk if users do not properly establish network connections. The following procedure provides the steps to properly establish a connection with JavaScript disabled. Network administrators are strongly advised to instruct users on how to properly establish connections using the procedure in this section.

Note Failure to follow this procedure can cause user credentials to be passed to a network web server other than the authentication proxy or can cause the authentication proxy to reject the login attempt.

To verify client connections using the authentication proxy with JavaScript disabled on the client browser, follow this procedure:

Step 1 Initiate an HTTP connection through the firewall.

This generates the authentication proxy login page.

Step 2 From the authentication proxy login page at the client, enter the username and password.

Step 3 Click **OK** to submit the username and password to the AAA server.

A popup window appears indicating whether the login attempt succeeded or failed. If the popup window indicates successful authentication, go to Step 7.

Step 4 If the popup window displays a failed authentication message, click **Close** on the browser **File** menu.

Note Do not click **Reload** (**Refresh** for Internet Explorer) to close the popup window.

Step 5 From the original authentication login page, click **Reload** (**Refresh** for Internet Explorer) on the browser toolbar. The user login credentials are cleared from the form.

Note Do not click **OK**. You must click **Reload** or **Refresh** to clear the username and password and to reload the form before attempting to log in again.

Step 6 Enter the username and password again.

If the authentication is successful, a window appears displaying a successful authentication message. If the window displays a failed authentication message, go to Step 4.

Step 7 Click **Close** on the browser **File** menu.

Step 8 From the original authentication proxy login page, click **Reload** (**Refresh** for Internet Explorer) on the browser toolbar.

The authentication proxy completes the authenticated connection with the web server.

Monitoring and Maintaining the Authentication Proxy

This section describes how to view dynamic access list entries and how to manually remove authentication entries. This section contains the following sections:

- Displaying Dynamic ACL Entries
- Deleting Authentication Proxy Cache Entries

Displaying Dynamic ACL Entries

You can display dynamic access list entries when they are in use. After an authentication proxy entry is cleared by you or by the idle timeout parameter, you can no longer display it. The number of matches displayed indicates the number of times the access list entry was hit.

To view dynamic access lists and any temporary access list entries that are currently established by the authentication proxy, use the **show ip access-lists** command in privileged EXEC mode:

Command	Purpose
`router# show ip access-lists`	Displays the standard and extended access lists configured on the firewall, including dynamic ACL entries.

Consider the following example where ACL 105 is applied inbound at the input interface where you configure authentication proxy. The initial display shows the contents of the ACLs prior to authentication. The second display shows the same displays after user authentication with the AAA server.

Note If NAT is configured, the **show ip access list** command might display the translated host IP address for the dynamic ACL entry. This depends on whether the ACL is applied on the interface where NAT is applied **inside** or **outside**. If the ACL is applied on the NAT outside interface, the translated address is displayed. If the ACL is applied on the NAT inside interface, the IP address of the host initiating the connection is displayed. The **show ip auth-proxy cache** command always displays the IP address of the host initiating the connection.

Initial ACL entries prior to the authentication proxy:

```
Router# show ip access-lists
      .
      .
      .
Extended IP access list 105
    deny tcp any any eq telnet
    deny udp any any
    permit tcp any any (28 matches)
    permit ip any any
```

The ACL entries following user authentication are shown in boldface type:

```
Router# show ip access-lists
     .
     .
     .
Extended IP access list 105
   permit tcp host 192.168.25.215 any eq 26
   permit icmp host 192.168.25.215 host 60.0.0.2
   permit tcp host 192.168.25.215 any eq telnet
   permit tcp host 192.168.25.215 any eq ftp
   permit tcp host 192.168.25.215 any eq ftp-data
   permit tcp host 192.168.25.215 any eq smtp
   deny tcp any any eq telnet
   deny udp any any
   permit tcp any any (76 matches)
   permit ip any any
```

Deleting Authentication Proxy Cache Entries

When the authentication proxy is in use, dynamic access lists dynamically grow and shrink as authentication entries are added and deleted. To display the list of authentication entries, use the **show ip auth-proxy cache** command. To manually delete an authentication entry, use the **clear ip auth-proxy cache** command in privileged EXEC mode:

Command	Purpose
`router#` **`clear ip auth-proxy cache`** `{* | ` *`host ip address`*`}`	Deletes authentication proxy entries from the firewall before they time out. Use an asterisk to delete all authentication cache entries. Enter a specific IP address to delete an entry for a single host.

Authentication Proxy Configuration Examples

Configuring the authentication proxy feature requires configuration changes on both the router and the AAA server. The following sections provide authentication proxy configuration examples:

- Authentication Proxy Configuration Example
- Authentication Proxy, IPSec, and CBAC Configuration Example
- Authentication Proxy, IPSec, NAT, and CBAC Configuration Example
- AAA Server User Profile Example

Throughout these examples, the "!" symbol indicates a comment line. Comment lines precede the configuration entries being described.

Authentication Proxy Configuration Example

The following examples highlight the specific authentication proxy configuration entries. These examples do not represent a complete router configuration. Complete router configurations using the authentication proxy are included later in this chapter.

This section has the following sections:

- AAA Configuration
- HTTP Server Configuration

- Authentication Proxy Configuration
- Interface Configuration

AAA Configuration

```
aaa new-model
aaa authentication login default group tacacs group radius
! Set up the aaa new model to use the authentication proxy.
aaa authorization auth-proxy default group tacacs group radius
! Define the AAA servers used by the router.
tacacs-server host 172.31.54.143
tacacs-server key cisco
radius-server host 172.31.54.143
radius-server key cisco
```

HTTP Server Configuration

```
! Enable the HTTP server on the router.
ip http server
! Set the HTTP server authentication method to AAA.
ip http authentication aaa
! Define standard access list 61 to deny any host.
access-list 61 deny any
! Use ACL 61 to deny connections from any host to the HTTP server.
ip http access-class 61
```

Authentication Proxy Configuration

```
! Set the global authentication proxy timeout value.
ip auth-proxy auth-cache-time 60
! Apply a name to the authentication proxy configuration rule.
ip auth-proxy name HQ_users http
```

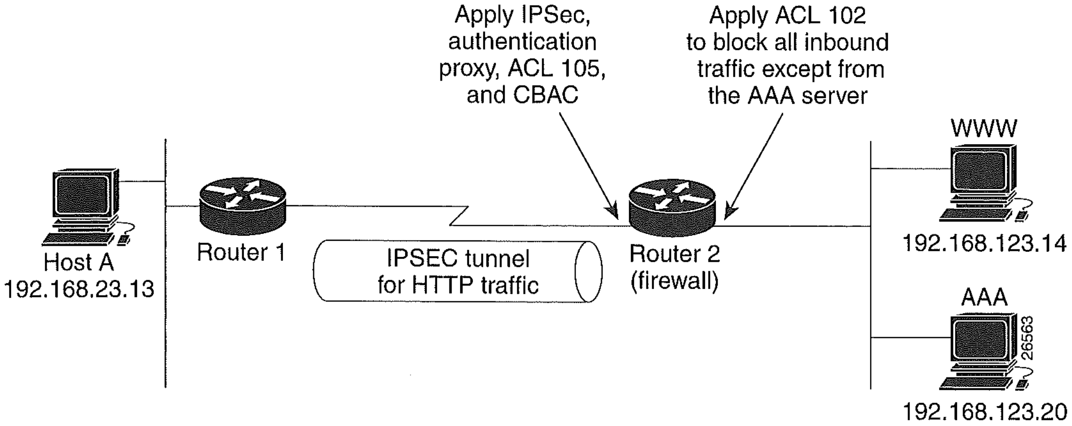
Interface Configuration

```
! Apply the authentication proxy rule at an interface.
interface e0
   ip address 10.1.1.210 255.255.255.0
   ip auth-proxy HQ_users
```

Authentication Proxy, IPSec, and CBAC Configuration Example

This example shows a router configuration with the authentication proxy, IPSec, and CBAC features. Figure 23 illustrates the configuration.

Figure 23 Authentication Proxy, IPSec, and CBAC Configuration Example


Apply IPSec, authentication proxy, ACL 105, and CBAC
Apply ACL 102 to block all inbound traffic except from the AAA server
WWW
Host A
192.168.23.13
Router 1
IPSEC tunnel for HTTP traffic
Router 2 (firewall)
192.168.123.14
AAA
26563
192.168.123.20

In this example, Host A initiates an HTTP connection with the web server (WWW). The HTTP traffic between Router 1 and Router 2 is encrypted using IPSec. The authentication proxy, IPSec, and CBAC are configured at interface Serial0 on Router 2, which is acting as the firewall. ACL 105 blocks all traffic at interface Serial0. ACL 102 is applied at interface Ethernet0 on Router 2 to block traffic on that interface, except traffic from the AAA server.

When Host A initiates an HTTP connection with the web server, the authentication proxy prompts the user at Host A for a username and password. These credentials are verified with the AAA server for authentication and authorization. If authentication is successful, the per-user ACLs are downloaded to the firewall to permit services.

The following examples provide both the Router 1 and Router 2 configurations for completeness.

- Router 1 Configuration
- Router 2 Configuration

Router 1 Configuration

```
! Configure Router 1 for IPSec.
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router1
!
logging buffered 4096 debugging
no logging console
enable secret 5 $1$E0OB$AQF1vFZM3fLr3LQAOsudL/
enable password junk
!
username Router2 password 0 welcome
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco1234 address 10.0.0.2
!
crypto ipsec transform-set rule_1 ah-sha-hmac esp-des esp-sha-hmac
!
 !
 crypto map testtag 10 ipsec-isakmp
 set peer 10.0.0.2
 set transform-set rule_1
 match address 155
!
interface Ethernet0/0
 ip address 192.168.23.2 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Serial3/1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 encapsulation PPP
 ip route-cache
 no ip mroute-cache
 no keepalive
 no fair-queue
 clockrate 56000
 crypto map testtag
!
!
ip classless
ip route 192.168.123.0 255.255.255.0 10.0.0.2
! Identify the IPSec specific traffic.
access-list 155 permit tcp host 192.168.23.13 host 192.168.123.14 eq www
access-list 155 permit tcp host 192.168.23.13 eq www host 192.168.123.14
```

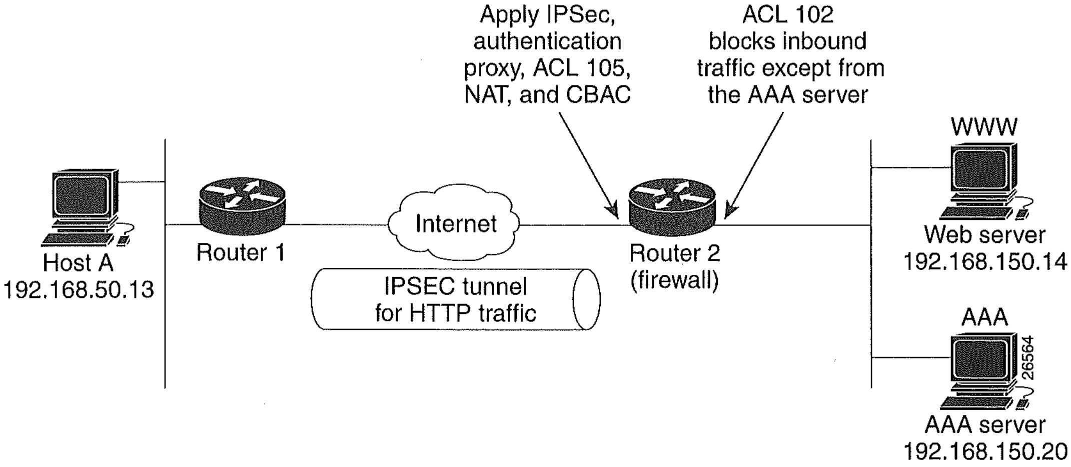
Router 2 Configuration

```
! Configure Router 2 as the firewall, using the authentication proxy, IPSec, and CBAC.
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router2
!
logging buffered 4096 debugging
aaa new-model
aaa authentication login default group tacacs
aaa authentication login console_line none
aaa authentication login special none
aaa authentication ppp default group tacacs
aaa authorization exec default group tacacs
! Configure AAA for the authentication proxy.
aaa authorization auth-proxy default group tacacs+
enable password junk
!
! Create the CBAC inspection rule HTTP_TEST.
ip inspect name rule22 http
ip inspect name rule22 tcp
ip inspect name rule22 ftp
ip inspect name rule22 smtp
!
! Create the authentication proxy rule PXY.
ip auth-proxy name pxy http
! Turn on display of the router name in the authentication proxy login page.
ip auth-proxy auth-proxy-banner
ip audit notify log
ip audit po max-events 100
!
! Configure IPSec.
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco1234 address 10.0.0.1
!
crypto ipsec transform-set rule_1 ah-sha-hmac esp-des esp-sha-hmac
!
 crypto map testtag 10 ipsec-isakmp
 set peer 10.0.0.1
 set transform-set rule_1
 match address 155
!
! Apply the CBAC inspection rule and the authentication proxy rule at interface
! Serial0/0.
interface Serial0/0
 ip address 10.0.0.2 255.0.0.0
 ip access-group 105 in
 no ip directed-broadcast
 ip inspect rule22 in
 ip auth-proxy pxy
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 no keepalive
 no fair-queue
crypto map testtag
!
interface Ethernet0/1
 ip address 192.168.123.2 255.255.255.0
 ip access-group 102 in
 no ip directed-broadcast
 ip route-cache
 no ip mroute-cache
!
no ip classless
ip route 192.168.23.0 255.255.255.0 10.0.0.1
ip route 192.168.50.0 255.255.255.0 16.0.0.1
! Configure the HTTP server.
ip http server
ip http access-class 15
ip http authentication aaa
!
! Create ACL 15 to block all traffic for the http server.
access-list 15 deny any
! Create ACL 102 to block all traffic inbound on interface Ethernet0/1 except for
! traffic from the AAA server.
access-list 102 permit tcp host 192.168.123.20 eq tacacs host 192.168.123.2
access-list 102 deny   tcp any any
access-list 102 deny   udp any any
access-list 102 permit ip any any
! Create ACL 105 to block all traffic inbound on interface Serial0/0. Permit only IP
! protocol traffic.
access-list 105 deny   tcp any any
access-list 105 deny   udp any any
access-list 105 permit ip any any
! Identify the IPSec specific traffic.
access-list 155 permit tcp host 192.168.123.14 host 192.168.23.13 eq www
access-list 155 permit tcp host 192.168.123.14 eq www host 192.168.23.13
!
! Define the AAA server host and encryption key.
tacacs-server host 192.168.123.14
tacacs-server key cisco
!
line con 0
 exec-timeout 0 0
 login authentication special
 transport input none
line aux 0
 transport input all
 speed 38400
 flowcontrol hardware
line vty 0 4
 password lab
```

Authentication Proxy, IPSec, NAT, and CBAC Configuration Example

This example provides a router configuration with the authentication proxy, IPSec, NAT, and CBAC features. Figure 24 illustrates the configuration.

Figure 24 Authentication Proxy, IPSec, and CBAC Configuration Example


Apply IPSec, authentication proxy, ACL 105, NAT, and CBAC
ACL 102 blocks inbound traffic except from the AAA server
WWW
Host A
192.168.50.13
Router 1
Internet
IPSEC tunnel for HTTP traffic
Router 2 (firewall)
Web server
192.168.150.14
AAA
AAA server
192.168.150.20
26564

In this example, Host A initiates an HTTP connection with the web server (WWW). The HTTP traffic between Router 1 (interface BRI0) and Router 2 (interface Serial2) is encrypted using IPSec. The authentication proxy is configured on Router 2, which is acting as the firewall. The authentication proxy, NAT, and CBAC are configured at interface Serial2, which is acting as the firewall. ACL 105 blocks all traffic at interface Serial2. ACL 102 is applied at interface Ethernet0 on Router 2 to block traffic on that interface, except traffic from the AAA server. In this example, the authentication proxy uses standard ACL 10 to specify the hosts using the authentication proxy feature.

When any host in ACL 10 initiates an HTTP connection with the web server, the authentication proxy prompts the user at that host for a username and password. These credentials are verified with AAA server for authentication and authorization. If authentication is successful, the per-user ACLs are downloaded to the firewall to permit services.

The following examples provide both the Router 1 and Router 2 configurations for completeness.

- Router 1 Configuration
- Router 2 Configuration

Router 1 Configuration

```
! Configure Router 1 for IPSec.
version 12.0
 service timestamps debug uptime
 service timestamps log uptime
 no service password-encryption
 !
 hostname Router1
 !
 logging buffered 4096 debugging
 no logging console
 !
isdn switch-type basic-5ess
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco1234 address 16.0.0.2
 crypto ipsec transform-set rule_1 ah-sha-hmac esp-des esp-sha-hmac
 !
  !
  crypto map testtag 10 ipsec-isakmp
  set peer 16.0.0.2
  set transform-set rule_1
  match address 155
 !
 !
 process-max-time 200
 !
 interface BRI0
  ip address 16.0.0.1 255.0.0.0
  no ip directed-broadcast
  encapsulation ppp
  dialer idle-timeout 5000
  dialer map ip 16.0.0.2 name router2 broadcast 50006
  dialer-group 1
  isdn switch-type basic-5ess
  crypto map testtag
 !
 interface FastEthernet0
  ip address 192.168.50.2 255.255.255.0
  no ip directed-broadcast
 !
 ip classless
 ip route 192.168.150.0 255.255.255.0 16.0.0.2
 no ip http server
! Identify the IPSec specific traffic.
 access-list 155 permit tcp host 192.168.50.13 host 192.168.150.100 eq www
 access-list 155 permit tcp host 192.168.50.13 eq www host 192.168.150.100
 dialer-list 1 protocol ip permit
 !
 line con 0
  exec-timeout 0 0
  transport input none
 line aux 0
 line vty 0 4
  password lab
  login
```

Router 2 Configuration

```
! Configure Router 2 as the firewall, using the authentication proxy, IPSec, NAT, and
! CBAC.
 version 12.0
 service timestamps debug uptime
 service timestamps log uptime
 no service password-encryption
 !
 hostname router2
 !
 logging buffered 4096 debugging
 aaa new-model
 aaa authentication login default group tacacs+
 aaa authentication login console_line none
 aaa authorization exec default group tacacs+
 ! Configure AAA for the authentication proxy.
 aaa authorization auth-proxy default group tacacs+
!
 ! Create the CBAC inspection rule "rule44."
 ip inspect name rule44 http java-list 5
 ip inspect name rule44 tcp
 ip inspect name rule44 ftp
 ip inspect name rule44 smtp
 !
 ! Create the authentication proxy rule "pxy." Set the timeout value for rule
 ! pxy to three minutes. Standard ACL 10 is applied to the rule.
 ip auth-proxy name pxy http list 10 auth-cache-time 3
 isdn switch-type primary-5ess
 !
 ! Configure IPSec.
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco1234 address 16.0.0.1
 !
 !
 crypto ipsec transform-set rule_1 ah-sha-hmac esp-des esp-sha-hmac
 !
  !
  crypto map testtag 10 ipsec-isakmp
  set peer 16.0.0.1
  set transform-set rule_1
  match address 155
!
 controller T1 2/0
  framing esf
  linecode b8zs
  pri-group timeslots 1-24
!
! Apply ACL 102 inbound at interface Ethernet0/1 and configure NAT.
interface Ethernet0/1
 ip address 192.168.150.2 255.255.255.0
 ip access-group 102 in
 no ip directed-broadcast
 ip nat inside
 no ip mroute-cache
!
! Apply the authentication proxy rule PXY, CBAC inspection rule HTTP_TEST, NAT, and
! and ACL 105 at interface Serial2/0:23.
interface Serial2/0:23
 ip address 16.0.0.2 255.0.0.0
 ip access-group 105 in
 no ip directed-broadcast
 ip nat outside
 ip inspect rule44 in
 ip auth-proxy pxy
 encapsulation ppp
 ip mroute-cache
 dialer idle-timeout 5000
 dialer map ip 16.0.0.1 name router1 broadcast 71011
 dialer-group 1
 isdn switch-type primary-5ess
 fair-queue 64 256 0
 crypto map testtag
!
! Use NAT to translate the Web server address.
ip nat inside source static 192.168.150.14 192.168.150.100
ip classless
ip route 192.168.50.0 255.255.255.0 16.0.0.1
! Configure the HTTP server.
ip http server
ip http access-class 15
ip http authentication aaa
!
! Create standard ACL 5 to specify the list of hosts from which to accept java applets.
! ACL 5 is used to block Java applets in the CBAC inspection rule named "rule44," which
! is applied at interface Serial2/0:23.
access-list 5 permit any
! Create standard ACL 10 to specify the hosts using the authentication proxy. This ACL
! used in the authentication proxy rule named "PXY", which is applied at interface
! Serial2/0:23.
access-list 10 permit any
! Create ACL 15 to block all traffic for the http server.
access-list 15 deny any
! Create extended ACL 102 to block all traffic inbound on interface Ethernet0/1
! except for traffic from the AAA server.
access-list 102 permit tcp host 192.168.150.20 eq tacacs 192.168.150.2
access-list 102 deny   tcp any any
access-list 102 deny   udp any any
access-list 102 permit ip any any
! Create extended ACL 105 to block all TCP and UDP traffic inbound on interface
! Serial2/0:23.
access-list 105 deny   tcp any any
access-list 105 deny   udp any any
access-list 105 permit ip any any
! Identify the IPSec specific traffic.
access-list 155 permit tcp host 192.168.150.100 host 192.168.50.13 eq www
access-list 155 permit tcp host 192.168.150.100 eq www host 192.168.50.13
dialer-list 1 protocol ip permit
! Define the AAA server host and encryption key.
tacacs-server host 192.168.126.14
tacacs-server key cisco
!
line con 0
 exec-timeout 0 0
! Define the AAA server host and encryption key.
 login authentication console_line
 transport input none
 line aux 0
 line vty 0 4
 password lab
!
!
 end
```

AAA Server User Profile Example

This section includes examples of the authentication proxy user profile entries on the AAA servers. The **proxyacl** entries define the user access privileges. After successfully using the authentication proxy to log in, these entries are transferred to the firewall router. Each entry in the profile must specify **permit** access for the service or application. The source address in each entry is set to "any," which is replaced with the IP address of the authenticating host when the profile is downloaded to the firewall. The privilege level must be set to 15 for all AAA users.

This section has the following sections:

- CiscoSecure ACS 2.3 for Windows NT
- CiscoSecure ACS 2.3 for UNIX
- TACACS+ Server
- Livingston Radius Server
- Ascend Radius Server

CiscoSecure ACS 2.3 for Windows NT

This section describes how to configure authentication proxy on CiscoSecure ACS 2.3 for Windows NT. For detailed information about CiscoSecure ACS, refer to the documentation for that product.

The following sample configuration is for the TACACS+ service of CiscoSecure ACS for Windows NT.

Step 1 Click the Interface Configuration icon and click **TACACS+** (Cisco).

(a) Scroll down to New Services.

(b) Add a new service, "auth-proxy" in the Service field. Leave the Protocol field empty.

(c) Select both the User and Group check boxes for the new service.

(d) Scoll down to Advance Configuration Options and check the Per-user Advance TACACS+ features.

(e) Click **Submit**.

Step 2 Click the Network Configuration icon.

(a) Click the Add Entry icon for Network Access Servers and fill in the Network Access Server Hostname, IP address, and the key (the key configured on the router) fields.

(b) Select TACACS+ (Cisco) for the Authenticate Using option.

(c) Click the Submit + Restart icon.

Step 3 Click the Group Setup icon.

(a) Select a user group from the drop-down menu.

(b) Select the Users in Group check box.

(c) Select a user from the user list.

(d) In the User Setup list, scroll down to TACACS+ Settings and select the "auth-proxy" check box.

(e) Select the Custom Attributes check box.

(f) Add the profiles entries (do not use single or double quotes around the entries) and set the privilege level to 15.

```
priv-lvl=15
proxyacl#1=permit tcp any any eq 26
proxyacl#2=permit icmp any host 60.0.0.2
proxyacl#3=permit tcp any any eq ftp
proxyacl#4=permit tcp any any eq ftp-data
proxyacl#5=permit tcp any any eq smtp
proxyacl#6=permit tcp any any eq telnet
```

(g) Click **Submit.**

Step 4 Click the User Setup icon.

(a) Click **List All Users.**

(b) Add a username.

(c) Scoll down to User Setup Password Authentication.

(d) Select SDI SecurID Token Card from the Password Authentication drop-down menu.

(e) Select the previous configured user group 1.

(f) Click **Submit.**

Step 5 Click Group Setup icon again.

(a) Select the user group 1.

(b) Click **Users in Group.**

(c) Click **Edit Settings.**

(d) Click the Submit + Restart icon to make sure the latest configuration is updated and sent to the AAA server.

CiscoSecure ACS 2.3 for UNIX

This section describes how to configure authentication proxy on CiscoSecure ACS 2.3 for UNIX. For detailed information regarding CiscoSecure ACS, refer to the documentation for that product.

To manage the CiscoSecure ACS using the Administrator program, you need a web browser that supports Java and JavaScript. You must enable Java in the browser application. You can start the Java-based CiscoSecure Administrator advanced configuration program from any of the CiscoSecure ACS Administrator web pages.

The following sample configuration procedure is for the TACACS+ service of CiscoSecure ACS 2.3 for UNIX.

Step 1 On the CiscoSecure ACS web menu bar of the CiscoSecure ACS web interface, click **Advanced** and then click **Advanced** again.

The Java-based CiscoSecure Administrator advanced configuration program appears. It might require a few minutes to load.

Step 2 In the CiscoSecure Administrator advanced configuration program, locate and deselect Browse in the Navigator pane of the tabbed Members page.

This displays the Create New Profile icon.

Step 3 In the Navigator pane, do one of the following:

- Locate and click the group to which the user will belong.
- If you do not want the user to belong to a group, click the [Root] folder icon.

Step 4 Click **Create Profile** to display the New Profile dialog box.

Step 5 Make sure the Group check box is cleared.

Step 6 Enter the name of the user you want to create and click **OK**. The new user appears in the tree.

Step 7 Click the icon for the group or user profile in the tree that is displayed in the Navigator pane of the tabbed Members page.

Step 8 If necessary, in the Profile pane, click the Profile icon to expand it.

A list or dialog box that contains attributes applicable to the selected profile or service appears in the window at the bottom right of the screen. The information in this window changes depending on what you have selected in the Profile pane.

Step 9 Click **Service-String**.

Step 10 Click **string**, enter **auth-proxy** in the text field, and click **Apply**.

Step 11 Select the **Option** menu.

Step 12 On the **Option** menu, click **Default Attributes**.

Step 13 Change the attribute from Deny to **Permit**.

Step 14 Click **Apply**.

Step 15 On the **Option** menu, click **Attribute** and enter the privilege level in the text field:

```
priv-lvl=15
```

Step 16 On the **Option** menu, click **Attribute** and enter the **proxyacl** entries in the text field:

```
proxyacl#1="permit tcp any any eq 26"
```

Repeat this step for each additional service or protocol to add:

```
proxyacl#2="permit icmp any host 60.0.0.2"
proxyacl#3="permit tcp any any eq ftp"
proxyacl#4="permit tcp any any eq ftp-data"
proxyacl#5="permit tcp any any eq smtp"
proxyacl#6="permit tcp any any eq telnet"
```

Step 17 When you have finished making all your changes, click **Submit**.

TACACS+ Server

```
default authorization = permit
key = cisco
user = Brian {
login = cleartext cisco
service = auth-proxy
   {
      priv-lvl=15
      proxyacl#1="permit tcp any any eq 26"
      proxyacl#2="permit icmp any host 60.0.0.2
      proxyacl#3="permit tcp any any eq ftp"
      proxyacl#4="permit tcp any any eq ftp-data"
      proxyacl#5="permit tcp any any eq smtp"
      proxyacl#6="permit tcp any any eq telnet"
   }
}
```

Livingston Radius Server

```
Bob Password = "cisco" User-Service-Type=Outbound-User
cisco-avpair = "auth-proxy:priv-lvl=15",
cisco-avpair = "auth-proxy:proxyacl#1=permit tcp any any eq 26",
cisco-avpair = "auth-proxy:proxyacl#2=permit icmp any host 60.0.0.2",
cisco-avpair = "auth-proxy:proxyacl#3=permit tcp any any eq ftp",
cisco-avpair = "auth-proxy:proxyacl#4=permit tcp any any eq ftp-data",
cisco-avpair = "auth-proxy:proxyacl#5=permit tcp any any eq smtp",
cisco-avpair = "auth-proxy:proxyacl#6=permit tcp any any eq telnet"
```

Ascend Radius Server

```
Alice Password = "cisco" User-Service = Dialout-Framed-User
cisco-avpair = "auth-proxy:priv-lvl=15",
cisco-avpair = "auth-proxy:proxyacl#1=permit tcp any any eq 26",
cisco-avpair = "auth-proxy:proxyacl#2=permit icmp any host 60.0.0.2",
cisco-avpair = "auth-proxy:proxyacl#3=permit tcp any any eq ftp",
cisco-avpair = "auth-proxy:proxyacl#4=permit tcp any any eq ftp-data",
cisco-avpair = "auth-proxy:proxyacl#5=permit tcp any any eq smtp",
cisco-avpair = "auth-proxy:proxyacl#6=permit tcp any any eq telnet"
```

Configuring Port to Application Mapping

This chapter describes the Cisco Secure Integrated Software Port to Application Mapping (PAM) feature. PAM enables CBAC-supported applications to be run on nonstandard ports. Using PAM, network administrators can customize access control for specific applications and services to meet the distinct needs of their networks.

For a complete description of the PAM commands in this chapter, refer to the "Port to Application Mapping Commands" chapter of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

In This Chapter

This chapter has the following sections:

- Port to Application Mapping Overview
- PAM Configuration Task List
- Monitoring and Maintaining PAM
- PAM Configuration Examples

Port to Application Mapping Overview

Port to Application Mapping (PAM) is a feature of the Cisco Secure Integrated Software feature set. PAM allows you to customize TCP or UDP port numbers for network services or applications. PAM uses this information to support network environments that run services using ports that are different from the registered or well-known ports associated with an application.

Using the port information, PAM establishes a table of default port-to-application mapping information at the firewall. The information in the PAM table enables Context-based Access Control (CBAC) supported services to run on non-standard ports. Previously, CBAC was limited to inspecting traffic using only the well-known or registered ports associated with an application. Now, PAM allows network administrators to customize network access control for specific applications and services.

PAM also supports host or subnet specific port mapping, which allows you to apply PAM to a single host or subnet using standard access control lists (ACLs). Host or subnet specific port mapping is done using standard ACLs.

This section has the following sections:

- How PAM Works
- System-Defined Port Mapping
- PAM and CBAC
- When to Use PAM

How PAM Works

PAM generates a table of information that identifies specific applications with specific TCP or UDP port information. When the firewall router first starts up, the PAM table is populated with system-defined mapping information. As you customize the mapping information, the PAM table is modified with the new information. The information in the PAM table serves as the default port mapping for traffic passing through the firewall.

PAM works with CBAC to identify the applications associated with various port numbers, including services running on non-standard ports, as it inspect traffic passing through the firewall. Previously, CBAC was limited to inspecting traffic using only the well-known or registered ports associated with an application.

Entries in the PAM table provide three types of mapping information:

- System-Defined Port Mapping
- User-Defined Port Mapping
- Host-Specific Port Mapping

System-Defined Port Mapping

PAM creates a table, or database, of system-defined mapping entries using the well-known or registered port mapping information set up during the system start-up. The system-defined entries comprise all the services supported by CBAC, which requires the system-defined mapping information to function properly. The system-defined mapping information cannot be deleted or changed; that is, you cannot map HTTP services to port 21 (FTP) or FTP services to port 80 (HTTP).

Note You can override the system-defined entries for specific hosts using the PAM host-specific option. Refer to the section "Host-Specific Port Mapping" in this chapter.

Table 22 lists the default system-defined services and applications in the PAM table.

Table 22 System-Defined Port Mapping

Application Name	Well-Known or Registered Port Number	Protocol Description
cuseeme	7648	CU-SeeMe Protocol
exec	512	Remote Process Execution
ftp	21	File Transfer Protocol (control port)
http	80	Hypertext Transfer Protocol
h323	1720	H.323 Protocol (for example, MS NetMeeting, Intel Video Phone)

Table 22 System-Defined Port Mapping (continued)

Application Name	Well-Known or Registered Port Number	Protocol Description
login	513	Remote login
mgcp	2427	Media Gateway Control Protocol
msrpc	135	Microsoft Remote Procedure Call
netshow	1755	Microsoft NetShow
real-audio-video	7070	RealAudio and RealVideo
rtsp	8559	Real Time Streaming Protocol
shell	514	Remote command
sip	5060	Session Initiation Protocol
smtp	25	Simple Mail Transfer Protocol
sqlnet	1521	SQL-NET
streamworks	1558	StreamWorks Protocol
sunrpc	111	SUN Remote Procedure Call
telnet	23	Telnet
tftp	69	Trivial File Transfer Protocol
vdolive	7000	VDOLive Protocol

This section has the following sections:

- User-Defined Port Mapping
- Host-Specific Port Mapping

User-Defined Port Mapping

Network services or applications that use non-standard ports require user-defined entries in the PAM table. For example, your network might run HTTP services on the non-standard port 8000 instead of on the system-defined default port (port 80). In this case, you can use PAM to map port 8000 with HTTP services. If HTTP services run on other ports, use PAM to create additional port mapping entries. After you define a port mapping, you can overwrite that entry at a later time by simply mapping that specific port with a different application.

Note If you try to map an application to a system-defined port, a message appears that warns you of a mapping conflict.

User-defined port mapping information can also specify a range of ports for an application by establishing a separate entry in the PAM table for each port number in the range.

User-defined entries are saved with the default mapping information when you save the router configuration.

Host-Specific Port Mapping

User-defined entries in the mapping table can include host-specific mapping information, which establishes port mapping information for specific hosts or subnets. In some environments, it might be necessary to override the default port mapping information for a specific host or subnet.

With host-specific port mapping, you can use the same port number for different services on different hosts. This means that you can map port 8000 with HTTP services for one host, while mapping port 8000 with Telnet services for another host.

Host-specific port mapping also allows you to apply PAM to a specific subnet when that subnet runs a service that uses a port number that is different from the port number defined in the default mapping information. For example, hosts on subnet 192.168.21.0 might run HTTP services on non-standard port 8000, while other traffic through the firewall uses the default port for HTTP services, which is port 80.

Host-specific port mapping allows you to override a system-defined entry in the PAM table. For example, if CBAC finds an entry in the PAM table that maps port 25 (the system-defined port for SMTP) with HTTP for a specific host, CBAC identifies port 25 as HTTP protocol traffic on that host.

Note If the host-specific port mapping information is the same as an existing system-defined or user-defined default entries, host-specific port changes have no effect.

PAM and CBAC

CBAC uses the information in the PAM table to identify a service or application from traffic flowing through the firewall. With PAM, CBAC can associate non-standard port numbers with specific protocols. For example, if you use PAM to map port 8000 with HTTP services, CBAC can determine that traffic using port 8000 is an HTTP application.

When to Use PAM

Here are a few examples of when you might want to use PAM:

- Use PAM to apply a non-standard port numbers for a service or application.
- Use PAM when a specific host or subnet uses a port number for an application that is different than the default port number established in the PAM table.
- Use PAM when different hosts use the same port number for different applications.

PAM Configuration Task List

See the following sections for PAM configuration tasks. Each task in the list indicates if it is optional or required:

- Configuring Standard ACLs (Optional)
- Configuring PAM (Required)
- Verifying PAM (Optional)

Configuring Standard ACLs

If you require PAM for a specific host or subnet, use the **access-list** (standard) command in global configuration mode to define an ACL:

Command	Purpose
Router(config)# **access-list** *access-list-number* **permit** *source* [*source-wildcard*]	(Optional) Creates a standard ACL that defines the specific host or subnet for host-specific PAM. For complete information on access-list command, refer to the *Cisco IOS IP and IP Routing Command Reference*.

Configuring PAM

To configure PAM, use the **ip port-map** command in global configuration mode:

Command	Purpose
Router(config)# **ip port-map** *appl_name* **port** *port_num* [**list** *acl_num*]	Establishes a port mapping entry using the TCP or UDP port number and the application name. (Optional) Use the list option to associate this port mapping to the specific hosts in the ACL. (PAM uses standard access lists only.) If an access list is included, the hosts defined in that ACL have the application *appl_name* running on port *port_num*.

Verifying PAM

To verify the port mapping information, enter the **show ip port-map** command in privileged EXEC mode and review the entries:

```
router# show ip port-map
```

This command displays all entries in the PAM table, including the system-defined entries.

For PAM configuration examples using the commands in this chapter, refer to the "PAM Configuration Examples" section at the end of this chapter.

Monitoring and Maintaining PAM

This section describes commands used to monitor and maintain PAM.

Command	Purpose
router # **show ip port-map** [*appl_name* | **port** *port_num*]	Displays the port mapping information, including the system-defined entries. Include the application name to display a list of entries by application. Include the port number to display the entries by port.
Router(config)# **no ip port-map** *appl_name* **port** *port_num* [**list** *acl_num*]	Use the **no** form of the **ip port-map** command to delete user-defined port mapping information. This command has no effect on the system-defined port mapping information.

PAM Configuration Examples

The following sections provide PAM configuration examples:

- Mapping an Application to a Non-Standard Port Example
- Mapping an Application with a Port Range Example

- Invalid Port Mapping Entry Example
- Mapping an Application to a Port for a Specific Host Example
- Mapping an Application to a Port for a Subnet Example
- Overriding a System-Defined Port Mapping Example
- Mapping Different Applications to the Same Port Example

Mapping an Application to a Non-Standard Port Example

In this example, non-standard port 8000 is established as the user-defined default port mapping for HTTP services:

```
ip port-map http port 8000
```

Mapping an Application with a Port Range Example

The following PAM entries establish a range of non-standard ports for HTTP services:

```
ip port-map http 8001
ip port-map http 8002
ip port-map http 8003
ip port-map http 8004
```

Invalid Port Mapping Entry Example

This example is not valid because it tries to establish port 21, which is the system-defined default port for FTP, as the user-defined port for HTTP services:

```
ip port-map http port 21
```

Mapping an Application to a Port for a Specific Host Example

In this example, a specific host uses port 8000 for FTP services. ACL 10 identifies the server address (192.168.32.43), while port 8000 is mapped with FTP services:

```
access-list 10 permit 192.168.32.43
ip port-map ftp port 8000 list 10
```

Mapping an Application to a Port for a Subnet Example

In this example, a specific subnet runs HTTP services on port 8080. ACL 50 identifies the subnet, while port 8080 is mapped with HTTP services:

```
access-list 50 permit 192.168.92.0 0.0.0.255
ip port-map http 8080 list 50
```

Overriding a System-Defined Port Mapping Example

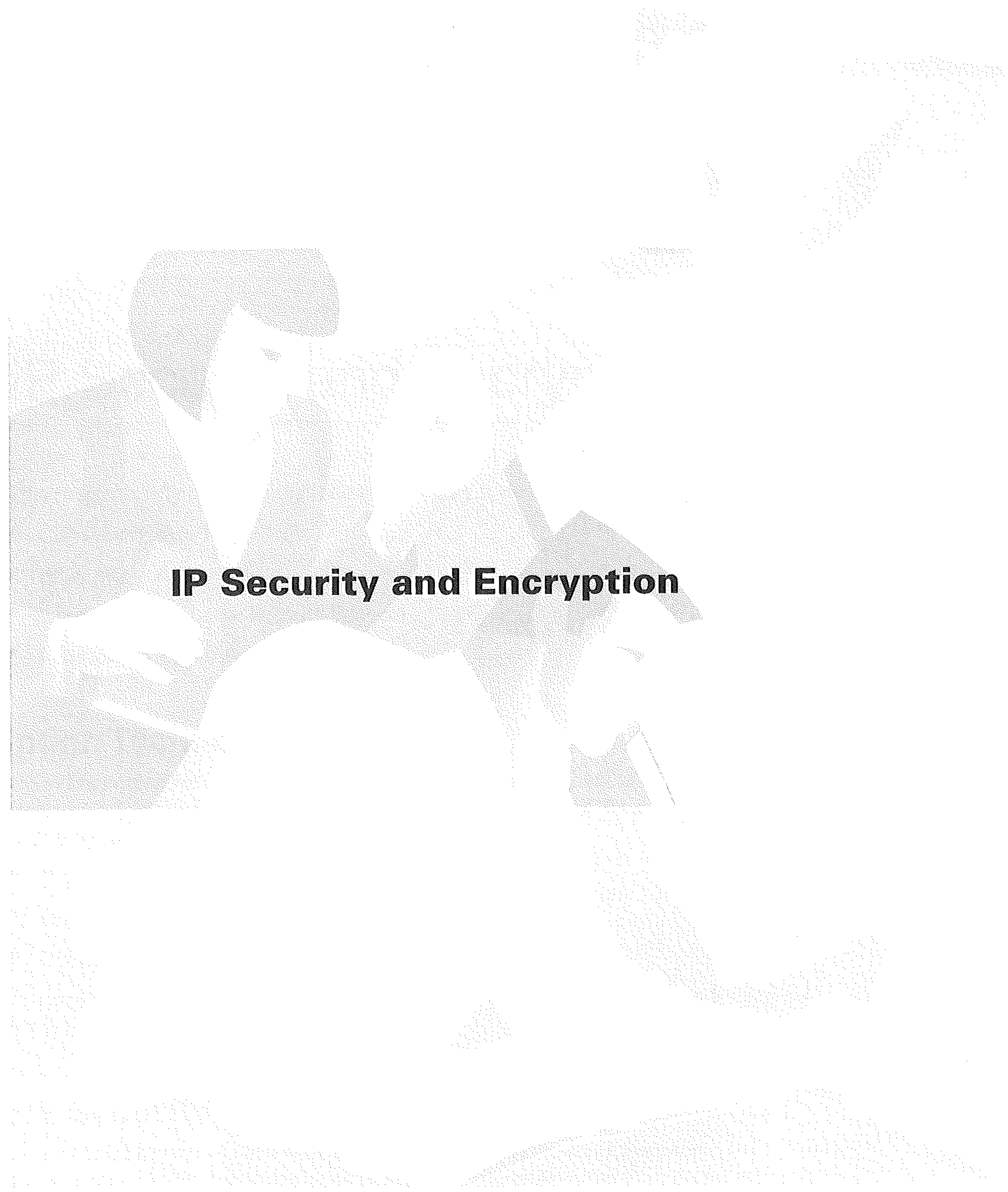
In this example, a specific host runs HTTP services on port 25, which is the system-defined port number for SMTP services. This requires a host-specific PAM entry that overrides the system-defined default port mapping for HTTP, which is port 80. ACL 15 identifies the host address (192.168.33.33), while port 25 is mapped with HTTP services:

```
access-list 15 permit 192.168.33.33
ip port-map http port 25 list 15
```

Mapping Different Applications to the Same Port Example

In this example, the same port number is required by different services running on different hosts. Port 8000 is required for HTTP services for host 192.168.3.4, while port 8000 is also required for FTP services for host 192.168.5.6. ACL 10 and ACL 20 identify the specific hosts, while the PAM entries map the ports with the services for each ACL:

```
access-list 10 permit 192.168.3.4
access-list 20 permit 192.168.5.6
ip port-map http port 8000 list 10
ip port-map http ftp 8000 list 20
```



IP Security and Encryption

IP Security and Encryption Overview

This chapter briefly describes the following security features and how they relate to each other:

- Cisco Encryption Technology
- IPSec Network Security
- Internet Key Exchange Security Protocol
- Certification Authority Interoperability

Cisco Encryption Technology

Cisco Encryption Technology (CET) is a proprietary security solution introduced in Cisco IOS Release 11.2. It provides network data encryption at the IP packet level and implements the following standards:

- Digital Signature Standard (DSS)
- Diffie-Hellman (DH) public key algorithm
- Data Encryption Standard (DES)

For more information regarding CET, refer to the chapter "Configuring Cisco Encryption Technology."

IPSec Network Security

IPSec is a framework of open standards developed by the Internet Engineering Task Force (IETF) that provides security for transmission of sensitive information over unprotected networks such as the Internet. It acts at the network level and implements the following standards:

- IPSec
- Internet Key Exchange (IKE)
- Data Encryption Standard (DES)
- MD5 (HMAC variant)
- SHA (HMAC variant)
- Authentication Header (AH)
- Encapsulating Security Payload (ESP)

IPSec services provide a robust security solution that is standards-based. IPSec also provides data authentication and anti-replay services in addition to data confidentiality services, while CET provides only data confidentiality services.

For more information regarding IPSec, refer to the chapter "Configuring IPSec Network Security."

IPSec Encryption Technology

IPSec shares the same benefitsc as CET: both technologies protect sensitive data that travels across unprotected networks, and, like CET, IPSec security services are provided at the network layer, so you do not have to configure individual workstations, PCs, or applications. This benefit can provide a great cost savings. Instead of providing the security services you do not need to deploy and coordinate security on a per-application, per-computer basis, you can simply change the network infrastructure to provide the needed security services.

IPSec encryption offers a number of additional benefits not present in CET:

- Because IPSec is standards-based, enables Cisco devices to interoperate with other IPSec-compliant networking devices to provide the IPSec security services. IPSec-compliant devices could include both Cisco devices and non-Cisco devices such as PCs, servers, and other computing systems.

  Cisco and its partners, including Microsoft, are planning to offer IPSec across a wide range of platforms, including Cisco IOS software, the Cisco PIX Firewall, and Window 2000.

- Enables a mobile user to establish a secure connection back to the office. For example, the user can establish an IPSec "tunnel" with a corporate firewall—requesting authentication services—in order to gain access to the corporate network; all of the traffic between the user and the firewall will then be authenticated. The user can then establish an additional IPSec tunnel—requesting data privacy services—with an internal router or end system.

- Provides support for the Internet Key Exchange (IKE) protocol and for digital certificates. IKE provides negotiation services and key derivation services for IPSec. Digital certificates allow devices to be automatically authenticated to each other without the manual key exchanges required by Cisco Encryption Technology. For more information, see the "Configuring Internet Key Exchange Security Protocol" chapter.

  This support allows IPSec solutions to scale better than CET solutions, making IPSec preferable in many cases for use with medium-sized, large-sized, and growing networks, where secure connections between many devices is required.

Differences Between IPSec and Cisco Encryption Technology

Should you implement CET or IPSec network security in your network? The answer depends on your requirements.

If you require only Cisco router-to-Cisco router encryption, then you could run CET, which is a more mature, higher-speed solution.

If you require a standards-based solution that provides multivendor interoperability or remote client connections, then you should implement IPSec. Also, if you want to implement data authentication with or without privacy (encryption), then IPSec is the right choice.

If you want, you can configure both CET and IPSec simultaneously in your network, even simultaneously on the same device. A Cisco device can simultaneously have CET secure sessions and IPSec secure sessions, with multiple peers.

Table 23 compares Cisco Encryption Technology to IPSec.

Table 23 Cisco Encryption Technology. vs. IPSec

Feature	Cisco Encryption Technology	IPSec
Availability	Cisco IOS Release 11.2 and later.	Cisco IOS Release 11.3(3)T and later.
Standards	Pre-IETF standards.	IETF standard.
Interoperability	Cisco router to Cisco router.	All IPSec compliant implementations.
Remote Access Solution	Not supported.	Client encryption will be supported.
Device Authentication	Manual between each peer at installation.	IKE uses digital certificates as a type of "digital ID card" (when Certification Authority support is configured); also supports manually-configured authentication shared secrets and manually-configured public keys.
Certificate Support	Not supported.	X509.V3 support; will support public key infrastructure standard when the standard is completed.
Protected Traffic	Selected IP traffic is encrypted according to extended access lists you define.	Selected IP traffic is encrypted and/or authenticated according to extended access lists; additionally, different traffic can be protected with different keys or different algorithms.
Hardware Support	Encryption Service Adapter (ESA) for the Cisco 7200/7500.	Integrated Services Adapter (ISA) for the Cisco 7200 and 7500; Integrated Services Module (ISM) for the Cisco 7100.
Packet Expansion	Not supported.	Tunnel mode adds a new IP and IPSec header to the packet; transport mode adds a new IPSec header.
Scope of Encryption	IP and ULP headers remain in the clear.	In tunnel mode, both the IP and ULP headers are encrypted; in transport mode, IP headers remain in the clear, but ULP headers are encrypted. (In tunnel mode, the inner IP header is also encrypted.)
Data Authentication with or without Encryption	Encryption only.	Can configure data authentication and encryption, or can use AH header to provide data authentication without encryption.
Internet Key Exchange (IKE) Support	Not supported.	Supported.
Redundant Topologies	Concurrent redundant Cisco Encryption Technology peers are not supported.	Concurrent redundant IPSec peers are supported

IPSec Performance Impacts

IPSec packet processing is slower than Cisco Encryption Technology packet processing for these reasons:

- IPSec offers per-packet data authentication, an additional task not performed with Cisco Encryption Technology.
- IPSec introduces packet expansion, which is more likely to require fragmentation/reassembly of IPSec packets.

IPSec Interoperability with Other Cisco IOS Software Features

You can use Cisco Encryption Technology and IPSec together; the two encryption technologies can coexist in your network. Each router may support concurrent encryption links using either IPSec or Cisco encryption technology. A single interface can even support the use of IPSec or CET for protecting different data flows.

Internet Key Exchange Security Protocol

Internet Key Exchange (IKE) security protocol is a key management protocol standard that is used in conjunction with the IPSec standard. IPSec can be configured without IKE, but IKE enhances IPSec by providing additional features, flexibility, and ease of configuration for the IPSec standard.

For more information regarding IKE, refer to the chapter "Configuring Internet Key Exchange Security Protocol."

Certification Authority Interoperability

Certification Authority (CA) interoperability is provided in support of the IPSec standard. It permits Cisco IOS devices and CAs to communicate so that your Cisco IOS device can obtain and use digital certificates from the CA. Although IPSec can be implemented in your network without the use of a CA, using a CA provides manageability and scalability for IPSec.

For more information regarding CA interoperability, refer to the chapter "Configuring Certification Authority Interoperability."

Configuring Cisco Encryption Technology

This chapter describes how to configure your router for network data encryption using Cisco Encryption Technology (CET).

In This Chapter

This chapter includes the following sections:

Note Whenever the term "encryption" is used in this chapter, it refers only to encryption of network data, not to other types of encryption.

For a complete description of the encryption commands in this chapter, refer to the "Cisco Encryption Technology Commands" chapter in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

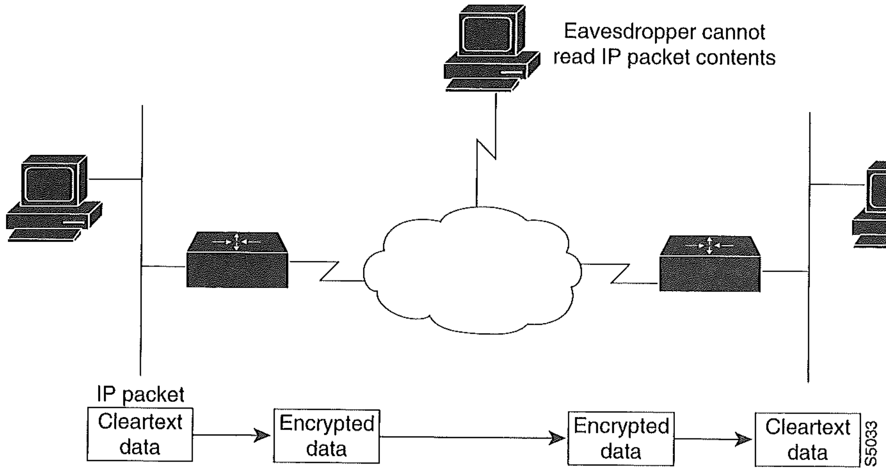
Why Encryption?

Data that traverses unsecured networks is open to many types of attacks. Data can be read, altered, or forged by anybody who has access to the route that your data takes. For example, a protocol analyzer can read packets and gain classified information. Or, a hostile party can tamper with packets and cause damage by hindering, reducing, or preventing network communications within your organization.

Encryption provides a means to safeguard network data that travels from one Cisco router to another across unsecured networks. Encryption is particularly important if classified, confidential, or critical data is being sent.

Figure 25 illustrates the encryption of an IP packet as it travels across an unsecured network.

Figure 25 IP Packet Encryption



Cisco's Implementation of Encryption

The following sections answer these questions:

- What Gets Encrypted?
- Where Are Packets Encrypted and Decrypted in the Network?
- When Can Encrypted Packets Be Exchanged?
- How Does an Encrypting Router Identify Other Peer Encrypting Routers?
- What Standards Are Implemented in Cisco's Encryption?
- How Does Cisco's Encryption Work?

What Gets Encrypted?

Network data encryption is provided at the IP packet level—only IP packets can be encrypted. (If you wish to encrypt a network protocol other than IP, you must encapsulate the protocol within an IP packet.)

An IP packet is encrypted/decrypted only if the packet meets criteria you establish when you configure a router for encryption.

When encrypted, individual IP packets can be detected during transmission, but the IP packet contents (payload) cannot be read. Specifically, the IP header and upper-layer protocol headers (for example, TCP or UDP) are not encrypted, but all payload data within the TCP or UDP packet will be encrypted, and therefore not readable during transmission.

Where Are Packets Encrypted and Decrypted in the Network?

The actual encryption and decryption of IP packets occur only at routers that you configure for CET. Such routers are considered to be *peer encrypting routers* (or simply *peer routers*). Intermediate hops do not participate in encryption/decryption.

Often, peer routers are situated at the edges of unsecured networks (such as the Internet), in order to provide secure communications between two secured networks that are physically separated. Cleartext (not encrypted) traffic that enters a peer router from the secure network side is encrypted and forwarded across the unsecure network. When the encrypted traffic reaches the remote peer router, the router decrypts the traffic before forwarding it into the remote secure network.

Packets are encrypted at one peer router's outbound interface and decrypted at the other peer router's inbound interface.

When Can Encrypted Packets Be Exchanged?

Encrypted packets can be exchanged between peer routers only during encrypted sessions. When a peer router detects a packet that should be encrypted, an encrypted session must first be established. After an encrypted session is established, encrypted traffic can pass freely between peer routers. When the session expires, a new session must be established before encrypted traffic can continue to be sent.

How Does an Encrypting Router Identify Other Peer Encrypting Routers?

During the setup of every encrypted session, both participating peer routers attempt to authenticate each other. If either authentication fails, the encrypted session will not be established, and no encrypted traffic will pass. Peer authentication ensures that only known, trusted peer routers exchange encrypted traffic, and prevents routers from being tricked into sending sensitive encrypted traffic to illegitimate or fraudulent destination routers.

What Standards Are Implemented in Cisco's Encryption?

To provide encryption services, Cisco implements the following standards: Digital Signature Standard (DSS), the Diffie-Hellman (DH) public key algorithm, and Data Encryption Standard (DES). DSS is used for peer router authentication. The DH algorithm and DES standard are used to initiate and conduct encrypted communication sessions between participating peer routers.

How Does Cisco's Encryption Work?

The following sections provide an overview of Cisco's encryption process:

- You Enable Peer Router Authentication with a DSS Key Exchange
- A Router Establishes an Encrypted Session with a Peer
- Peer Routers Encrypt and Decrypt Data During an Encrypted Session

You Enable Peer Router Authentication with a DSS Key Exchange

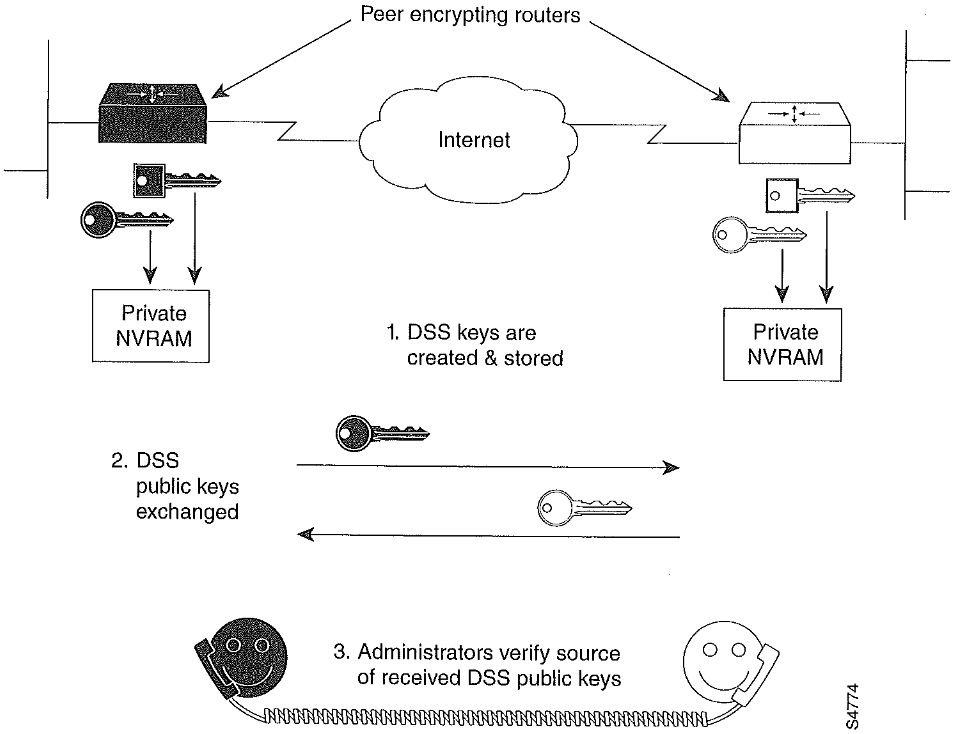
Peer router authentication occurs during the setup of each encrypted session. But before peer routers can authenticate each other, you must generate Digital Signature Standard (DSS) keys (both public and private DSS keys) for each peer, and you must exchange (and verify) the DSS public keys with each peer (see Figure 26). You generate and exchange DSS keys only once per peer, and afterwards these DSS keys will be used each time an encrypted session occurs. (Generating and exchanging DSS keys are described later in the "Configuring Encryption" section.)

Each peer router's DSS keys are unique: a unique DSS public key, and a unique DSS private key. DSS private keys are stored in a private portion of the router's NVRAM, which cannot be viewed with commands such as **more system:running-config** or **more nvram:startup-config**. If you have a router with an Encryption Service Adapter (ESA), DSS keys are stored in the tamper resistant memory of the ESA.

The DSS private key is not shared with any other device. However, the DSS public key is distributed to all other peer routers. You must cooperate with the peer router's administrator to exchange public keys between the two peer routers, and you and the other administrator must verbally verify to each other the public key of the other router. (The verbal verification is sometimes referred to as "voice authentication.")

When an encrypted session is being established, each router uses the peer's DSS public key to authenticate the peer. The process of authenticating peers and establishing encrypted sessions is described next.

Figure 26 Exchanging DSS Keys (Overview)


Peer encrypting routers
Internet
Private NVRAM
1. DSS keys are created & stored
Private NVRAM
2. DSS public keys exchanged
3. Administrators verify source of received DSS public keys
S4774

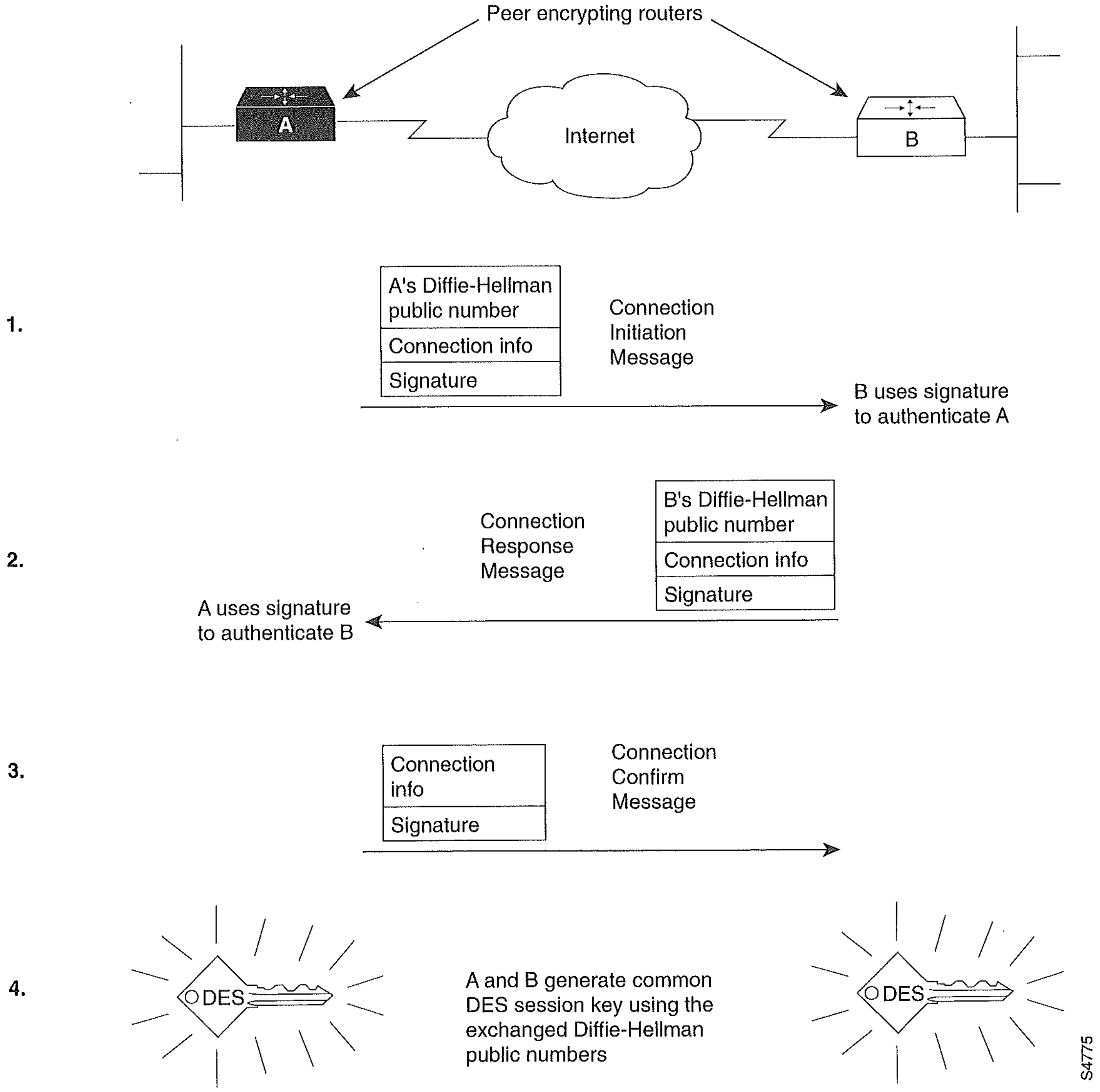
A Router Establishes an Encrypted Session with a Peer

An encrypted session must be established before a Cisco router can send encrypted data to a peer router. (See Figure 27.) An encrypted session is established whenever a router detects an IP packet that should be encrypted and no encrypted session already exists.

To establish a session, two peer routers exchange connection messages. These messages have two purposes. The first purpose is to authenticate each router to the other. Authentication is accomplished by attaching "signatures" to the connection messages: A signature is a character string that is created by each local router using its own DSS private key, and verified by the remote router using the local router's DSS public key (previously exchanged). A signature is always unique to the sending router and cannot be forged by any other device. When a signature is verified, the router that sent the signature is authenticated.

The second purpose of the connection messages is to generate a temporary DES key ("session key"), which is the key that will be used to actually encrypt data during the encrypted session. To generate the DES key, Diffie-Hellman (DH) numbers must be exchanged in the connection messages. Then, the DH numbers are used to compute a common DES session key that is shared by both routers.

Figure 27 Establishing an Encrypted Session

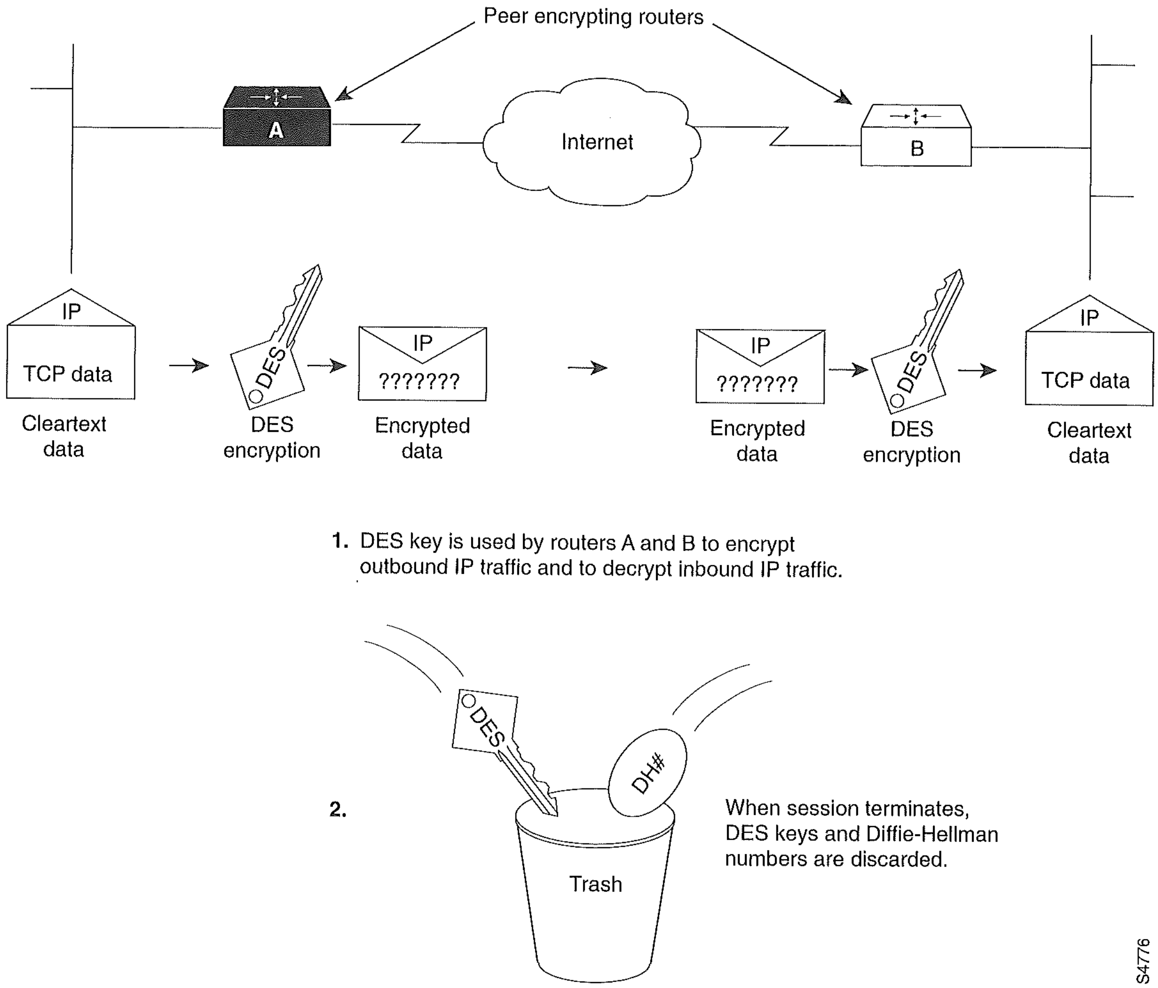

Peer encrypting routers
A
Internet
B
1.
A's Diffie-Hellman public number
Connection info
Signature
Connection Initiation Message
B uses signature to authenticate A
2.
Connection Response Message
B's Diffie-Hellman public number
Connection info
Signature
A uses signature to authenticate B
3.
Connection info
Signature
Connection Confirm Message
4.
DES
A and B generate common DES session key using the exchanged Diffie-Hellman public numbers
DES
S4775

Peer Routers Encrypt and Decrypt Data During an Encrypted Session

After both peer routers are authenticated and the session key (DES key) has been generated, data can be encrypted and transmitted. A DES encryption algorithm is used with the DES key to encrypt and decrypt IP packets during the encrypted session. (See Figure 28.)

An encrypted communication session will terminate when the session times out. When the session terminates, both the DH numbers and the DES key are discarded. When another encrypted session is required, new DH numbers and DES keys will be generated.

Figure 28 Encrypting Data



Additional Sources of Information

The following reading material can provide additional background information about network data encryption, including theory, standards, and legal requirements.

Applied Cryptography, Bruce Schneier

Network Security: Private Communication in a Public World, Kaufman, Perlman, and Specinen

Actually Useful Internet Security Techniques, Larry J. Hughes, Jr.

FIPS140, Federal Information Processing Standard

Defense Trade Regulations (Parts 120 to 126)

Information Security and Privacy in Network Environments, Office of Technology Assessment

Prework: Before Configuring Encryption

You should understand and follow the guidelines in this section *before* attempting to configure your system for CET. This section describes the following guidelines:

- Identifying Peer Routers
- Considering Your Network Topology
- Identifying Crypto Engines Within Each Peer Router
- Understanding Implementation Issues and Limitations

Identifying Peer Routers

You must identify all peer routers that will be participating in encryption. Peer routers are routers configured for encryption, between which all encrypted traffic is passed. These peers are usually routers within your administrative control that will be passing IP packets over an uncontrolled network (such as the Internet). Participating peer routers might also include routers not within your administrative control; however, this should only be the case if you share a trusted, cooperative relationship with the other router's administrator. This person should be known and trusted by both you and your organization.

Peer routers should be located within a network topology per the guidelines listed next.

Considering Your Network Topology

Take care in choosing a network topology between peer encrypting routers. Particularly, you should set up the network so that a stream of IP packets must use exactly one pair of encrypting routers at a time. Do not nest levels of encrypting routers. (That is, do not put encrypting routers in between two peer encrypting routers.)

Frequent route changes between pairs of peer encrypting routers, including for purposes of load balancing, will cause excessive numbers of connections to be set up and very few data packets to be delivered. Note that load balancing can still be used, but only if done transparently to the encrypting peer routers. That is, peer routers should not participate in the load balancing: only devices in between the peer routers should provide load balancing.

A common network topology used for encryption is a hub-and-spoke arrangement between an enterprise router and branch routers. Also, Internet firewall routers are often designated as peer encrypting routers.

Identifying Crypto Engines Within Each Peer Router

Encryption is provided by a software service called a *crypto engine*. To perform encryption at a router, you must first configure the router's crypto engine to be an encrypting peer; then you can configure any interface governed by that crypto engine to perform encryption. (To configure a crypto engine, you must at a minimum generate and exchange DSS keys for that engine, as described later in the "Configuring Encryption" section.)

Depending on your hardware configuration, different crypto engines will govern different router interfaces. In some instances, you may even need to configure multiple crypto engines as peers within a single router, particularly if a router has multiple interfaces that you want to use for encryption, and those interfaces are governed by different crypto engines.

There are three types of crypto engines—the Cisco IOS crypto engine, the VIP2 crypto engine, and the ESA crypto engine.

If you have a Cisco 7200, RSP7000, or 7500 series router with one or more VIP2 boards (VIP2-40 or higher) or ESA cards, your router can have multiple crypto engines. All other routers have only one crypto engine, the Cisco IOS crypto engine.

When you configure a crypto engine on a Cisco 7200, RSP7000, or 7500 series router, you must identify which engine you are configuring by specifying the engine's chassis slot number when you enter the crypto commands.

Note In Cisco 7500 and RSP7000-equipped Cisco 7000 systems, the ESA requires a VIP2-40 for operation and it must be installed in PA slot 1.

The three different crypto engines are described next.

The Cisco IOS Crypto Engine

Every router with Cisco IOS encryption software has a Cisco IOS crypto engine. For many Cisco routers, the Cisco IOS crypto engine is the only crypto engine available. The only exceptions are the Cisco 7200, RSP7000, and 7500 series routers, which can also have additional crypto engines as described in the next two sections.

If a router has no additional crypto engines, the Cisco IOS crypto engine governs all the router interfaces: you must configure the Cisco IOS crypto engine before you can configure any router interface for encryption.

The Cisco IOS crypto engine is identified by the chassis slot number of the Route Switch Processor (RSP). (For routers with no RSP, the Cisco IOS crypto engine is selected by default and does not need to be specifically identified during configuration.)

The VIP2 Crypto Engine (Cisco RSP7000 and 7500 Series Routers Only)

Cisco RSP7000 and 7500 series routers with a second-generation Versatile Interface Processor (VIP2) (version VIP2-40 or greater) have two crypto engines: the Cisco IOS crypto engine and the VIP2 crypto engine.

The VIP2 crypto engine governs the adjoining VIP2 port interfaces. The Cisco IOS crypto engine governs all remaining router interfaces. (These rules assume there is no ESA installed in the VIP2. If the VIP2 has an installed ESA, the interfaces are governed differently, as explained in the next section.)

The VIP2 crypto engine is identified by the chassis slot number of the VIP2.

Note In Cisco 7500 and RSP7000-equipped Cisco 7000 systems, the ESA requires a VIP2-40 for operation and it must be installed in PA slot 1.

The Encryption Service Adapter Crypto Engine (Cisco 7200, RSP7000, and 7500 Series Routers Only)

Cisco 7200, RSP7000, and 7500 series routers with an Encryption Service Adapter (ESA) have an ESA crypto engine.

Cisco 7200 Series Routers with an ESA

When a Cisco 7200 router has an active ESA, the ESA crypto engine—not the Cisco IOS crypto engine—governs all the router interfaces. (With an inactive ESA, the Cisco IOS crypto engine governs all the router interfaces. On the Cisco 7200, you can select which engine is active; only one engine is active at a time.)

The ESA plugs into the Cisco 7200 chassis, and the ESA crypto engine is identified by the ESA's chassis slot number.

Cisco RSP7000 and 7500 Series Routers with an ESA

The ESA and an adjoining port adapter plug into a VIP2 board. The ESA crypto engine—not the VIP2 crypto engine—governs the adjoining VIP2 port interfaces. The Cisco IOS crypto engine governs all remaining interfaces.

In a Cisco RSP7000 or 7500 series router, the ESA crypto engine is identified by the chassis slot number of the VIP2.

Understanding Implementation Issues and Limitations

Please note the following issues and limitations of encryption, described in this section:

- Encapsulation
- Multicast of Encrypted Traffic
- IP Fragmentation
- Restrictions for Switching Types with the VIP2
- Number of Simultaneous Encrypted Sessions
- Performance Impacts

Encapsulation

You can use any type of encapsulation with IP encryption, except as follows: If you have a second-generation Versatile Interface Processor (VIP2) with a serial interface, encryption will not work for traffic on the serial interface unless you use the Point-to-Point Protocol (PPP), High-Level Data Link Control (HDLC) protocol, or Frame Relay protocol. For example, you cannot use encryption if you have X.25 or SMDS configured for the serial interface of a VIP2.

Table 24 shows port adapter support by platform.

Table 24 Port Adapter Support

Interface	Encapsulation	7200 Software	7200 ESA	7500/VIP Distribution Software	7500/VIP ESA
4E, 8E, 5EFL	—	Yes	Yes	Yes	Yes
FE	—	Yes	Yes	Yes	Yes
4R	—	Yes	Yes	Yes	Yes
FDDI	—	Yes	Yes	Yes	Yes
100VG	—	Yes	Yes	Yes	Yes
4T	PPP, HDLC, Frame Relay	No	No	Yes	No
4T+, 8T	PPP	Yes	Yes	Yes	Yes
	HDLC	Yes	Yes	Yes	Yes
	Frame Relay	Yes	Yes	Yes	Yes
	X.25	Yes	Yes	No	No
	SMDS	Yes	Yes	No	No
HSSI	PPP	Yes	Yes	Yes	Yes
	HDLC	Yes	Yes	Yes	Yes
	Frame Relay	Yes	Yes	Yes	Yes
	X.25	Yes	Yes	No	No
	SMDS	Yes	Yes	No	No
CT1, CE1	PPP	Yes	Yes	Yes	Yes
	HDLC	Yes	Yes	Yes	Yes
	Frame Relay	Yes	Yes	Yes	Yes
	X.25	Yes	Yes	No	No
	SMDS	Yes	Yes	No	No
PRI	HDLC	Yes	Yes	No	No
	PPP	Yes	Yes	No	No
BRI	HDLC	Yes	Yes	No	No
	PPP	Yes	Yes	No	No
ATM	—	Yes	Yes	Yes	No
CT3	—	No	No	Yes	No

Multicast of Encrypted Traffic

Encrypted multicast is not supported.

IP Fragmentation

IP fragmentation is supported with encryption for all platforms except the VIP2. If you configure encryption for VIP2 interfaces, all IP fragments will be dropped.

Restrictions for Switching Types with the VIP2

If you configure encryption for VIP2 interfaces on a Cisco RSP7000 or 7500 series router, you must use distributed switching (DSW) on the source and destination encrypting/decrypting interfaces.

This restriction means that any protocol that is not compatible with DSW, such as SMDS, cannot be used on VIP2 encrypting interfaces.

Number of Simultaneous Encrypted Sessions

Each encrypting router can set up encrypted sessions with many other routers, if these are peer encrypting routers. Encrypting routers can also set up multiple simultaneous encrypted sessions with multiple peer routers. Up to 299 concurrent encrypted sessions per router can be supported.

Performance Impacts

Because of the high amount of processing required for encryption, if you use encryption heavily there will be performance impacts such as interface congestion or slowed CPU functioning. Using an ESA crypto engine rather than the Cisco IOS crypto engine can improve overall router performance because the Cisco IOS software will not be impacted by encryption processing.

Configuring Encryption

To pass encrypted traffic between two routers, you must configure encryption at both routers. This section describes the tasks required to configure encryption on one router: you must repeat these tasks for each peer encrypting router (routers that will participate in encryption).

To configure encryption on a router, complete the tasks described in the following sections:

- Generating DSS Public/Private Keys (required to configure a crypto engine)
- Exchanging DSS Public Keys (required to configure a crypto engine)
- Enabling DES Encryption Algorithms (required to configure the router)
- Defining Crypto Maps and Assigning Them to Interfaces (required to configure router interfaces)
- Backing Up Your Configuration

Note There are additional steps required if you configure encryption with GRE tunnels or if you configure encryption with an Encryption Service Adaptor (ESA). These additional steps are described later in this chapter, in the sections "Configuring Encryption with GRE Tunnels," "Configuring Encryption with an ESA in a VIP2," and "Configuring Encryption with an ESA in a Cisco 7200 Series Router." Before you configure encryption, refer to these other sections as appropriate.

For examples of the configurations in this section, see the section "Encryption Configuration Examples" at the end of this chapter.

Generating DSS Public/Private Keys

You must generate DSS keys for each crypto engine you will use. If you will use more than one crypto engine, you must generate DSS keys separately for each engine. (These are the crypto engines you previously identified per the description in the earlier section "Identifying Crypto Engines Within Each Peer Router.")

The DSS key pair that you generate is used by peer routers to authenticate each other before each encrypted session. The same DSS key pair is used by a crypto engine with all its encrypted sessions (regardless of the peer encrypting router that it connects to).

Generate DSS keys for a crypto engine by using at least the first of the following commands in global configuration mode:

Command	Purpose
`crypto key generate dss` *`key-name`* `[`*`slot`*`]`	Generates DSS public and private keys.
`show crypto key mypubkey dss [`*`slot`*`]`	Displays DSS public key (private key not viewable).
`copy system:running-config nvram:startup-config`	Saves DSS keys to private NVRAM. (Complete this task only for Cisco IOS crypto engines.)

Note You must perform the **copy system:running-config nvram:startup-config** (previously **copy running-config startup-config**) command to save Cisco IOS crypto engine DSS keys to a private portion of NVRAM. DSS keys are *not* saved with your configuration when you perform a **copy system:running-config rcp:** or **copy system:running-config tftp:** command.

If you are generating keys for an ESA crypto engine, the following occurs during DSS key generation:

- You are prompted to enter a password.
  - If you previously used the **crypto key zeroize dss** command to reset the ESA, you should create a new password for the ESA at this time.
  - If you previously used the **crypto card clear-latch** command to reset the ESA, you should now use the password you assigned when you reset the ESA. If you do not remember the password, you must clear the ESA with the **crypto key zeroize dss** command; you can then generate keys and create a new password for the ESA.
- The DSS keys are automatically saved to the tamper resistant memory of the ESA.

Configuring encryption with an ESA is described later in the sections "Configuring Encryption with an ESA in a VIP2" and "Configuring Encryption with an ESA in a Cisco 7200 Series Router."

Exchanging DSS Public Keys

You must exchange DSS public keys with all participating peer routers. This allows peer routers to authenticate each other at the start of encrypted communication sessions.

If your network contains several peer encrypting routers, you need to exchange DSS keys multiple times (once for each peer router). If you ever add an encrypting peer router to your network topology, you will then need to exchange DSS keys with the new router to enable encryption to occur with that new router.

Note When you exchange DSS keys, you must make a phone call to the administrator of the peer encrypting router. You need to be in voice contact with the other administrator during the key exchange in order to voice authenticate the source of exchanged DSS public keys.

You must exchange the DSS public keys of each crypto engine that you will use.

To successfully exchange DSS public keys, you must cooperate with a trusted administrator of the other peer router. You and the administrator of the peer router must complete the following steps in the order given (refer to Figure 29):

Step 1 Call the other administrator on the phone. Remain on the phone with this person until you complete all the steps in this list.

Step 2 You and the other administrator decide which of you will be called "PASSIVE" and which will be called "ACTIVE."

Step 3 PASSIVE enables a DSS exchange connection by using the following command in global configuration mode:

Command	Purpose
`crypto key exchange dss passive [tcp-port]`	Enables a DSS exchange connection.

Step 4 ACTIVE initiates a DSS exchange connection and sends a DSS public key by using the following command in global configuration mode:

Command	Purpose
crypto key exchange dss *ip-address key-name* [*tcp-port*]	Initiates a connection and send DSS public key.

The serial number and Fingerprint of ACTIVE's DSS public key will display on both of your screens. The serial number and Fingerprint are numeric values generated from ACTIVE's DSS public key.

Step 5 You both verbally verify that the serial number is the same on both your screens, and that the Fingerprint is the same on both your screens.

Step 6 If the displayed serial numbers and Fingerprints match, PASSIVE should agree to accept ACTIVE's DSS key by typing `y` at the prompt.

Step 7 PASSIVE sends ACTIVE a DSS public key by pressing <Return> at the screen prompt and selecting a crypto engine at the next prompt.

Step 8 PASSIVE's DSS serial number and Fingerprint display on both of your screens.

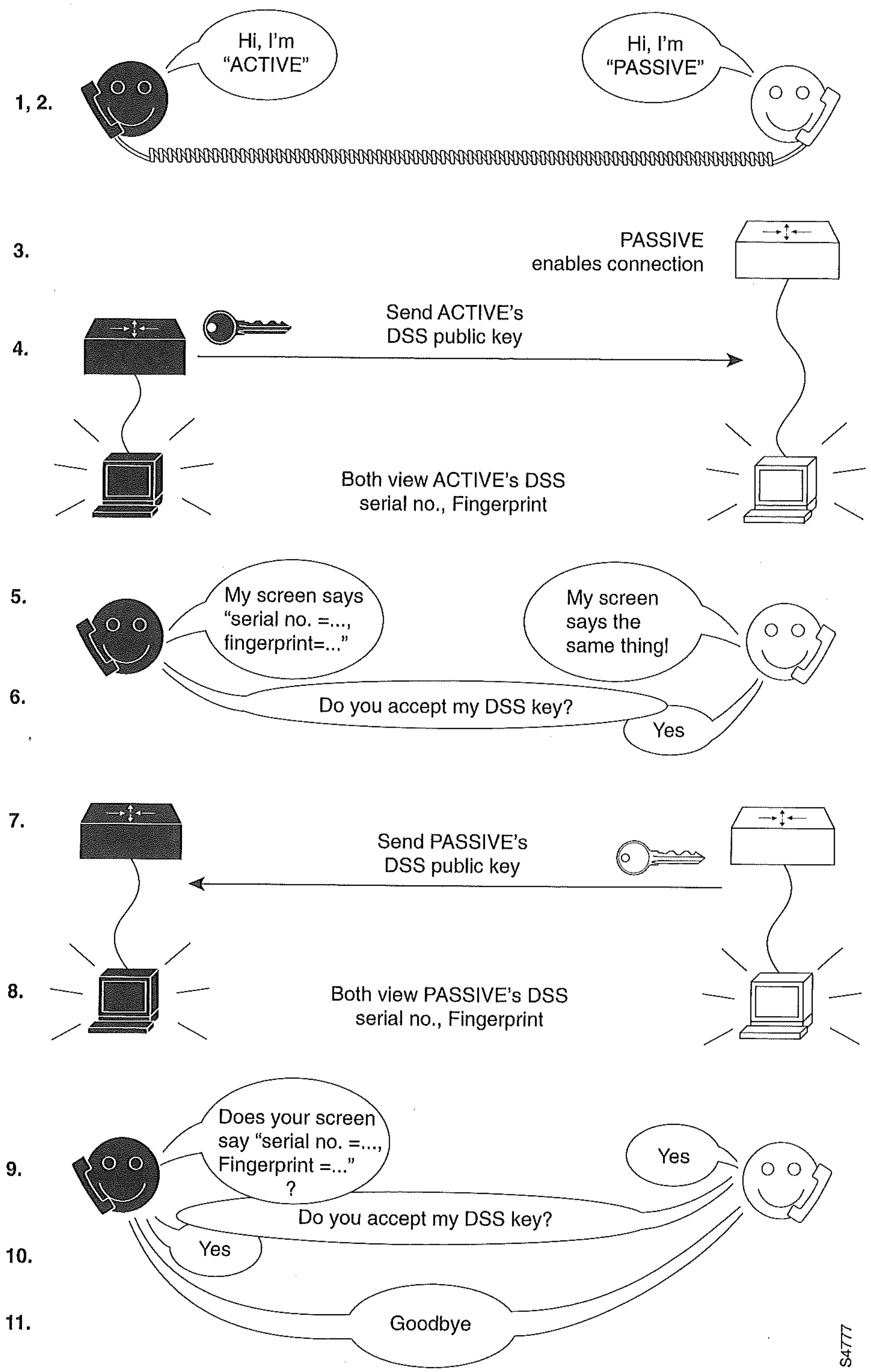
Step 9 As before, you both verbally verify that the PASSIVE's DSS serial number and Fingerprint match on your two screens.

Step 10 ACTIVE agrees to accept PASSIVE's DSS public key.

Step 11 The exchange is complete, and you can end the phone call.

The previous steps (illustrated in Figure 29) must be accomplished between your router and a peer router for every peer router with which you will be conducting encrypted sessions.

Figure 29 Exchanging DSS Public Keys


1, 2.
Hi, I'm "ACTIVE"
Hi, I'm "PASSIVE"
3.
PASSIVE enables connection
4.
Send ACTIVE's DSS public key
Both view ACTIVE's DSS serial no., Fingerprint
5.
My screen says "serial no. =..., fingerprint=..."
My screen says the same thing!
6.
Do you accept my DSS key?
Yes
7.
Send PASSIVE's DSS public key
8.
Both view PASSIVE's DSS serial no., Fingerprint
9.
Does your screen say "serial no. =..., Fingerprint =..." ?
Yes
10.
Do you accept my DSS key?
Yes
11.
Goodbye
S4777

Enabling DES Encryption Algorithms

Cisco routers use Data Encryption Standard (DES) encryption algorithms and DES keys to encrypt and decrypt data. You must globally enable (turn on) all the DES encryption algorithms that your router will use during encrypted sessions. If a DES algorithm is not enabled globally, you will not be able to use it. (Enabling a DES algorithm once allows it to be used by all crypto engines of a router.)

To conduct an encrypted session with a peer router, you must enable at least one DES algorithm that the peer router also has enabled. You must configure the same DES algorithm on both peer routers for encryption to work.

CET supports the following four types of DES encryption algorithms:

- DES with 8-bit Cipher FeedBack (CFB)
- DES with 64-bit CFB
- 40-bit variation of DES with 8-bit CFB
- 40-bit variation of DES with 64-bit CFB

The 40-bit variations use a 40-bit DES key, which is easier for attackers to "crack" than basic DES which uses a 56-bit DES key. However, some international applications might require you to use 40-bit DES because of export laws. Also, 8-bit CFB is more commonly used than 64-bit CFB, but requires more CPU time to process. Other conditions might also exist that require you to use one or another type of DES.

Note If you are running an exportable image, you can only enable and use 40-bit variations of DES. You cannot enable or use the basic DES algorithms, which are not available with exportable images.

One DES algorithm is enabled for your router by default. If you do not plan to use the default DES algorithm, you may choose to disable it. If you are running a non-exportable image, the DES default algorithm will be basic DES with 64-bit CFB. If you are running an exportable image, the DES default algorithm will be the 40-bit variation of DES with 64-bit CFB.

If you do not know if your image is exportable or non-exportable, you can use the **show crypto cisco algorithms** command to determine which DES algorithms are currently enabled.

Globally enable one or more DES algorithms by using one or more of the following commands in global configuration mode:

Command	Purpose
`crypto cisco algorithm des [cfb-8	cfb-64]`	Enables DES with 8-bit or 64-bit CFB.
`crypto cisco algorithm 40-bit-des [cfb-8	cfb-64]`	Enables 40-bit DES with 8-bit or 64-bit CFB.
`show crypto cisco algorithms`	Displays all enabled DES algorithms.

Defining Crypto Maps and Assigning Them to Interfaces

The purpose of this task is to tell your router which interfaces should encrypt/decrypt traffic, which IP packets to encrypt or decrypt at those interfaces, and also which DES encryption algorithm to use when encrypting/decrypting the packets.

There are actually three steps required to complete this task:

Step 1 Setting Up Encryption Access List (to be used in the crypto map definition)

Step 2 Defining Crypto Maps

Step 3 Applying Crypto Maps to Interfaces

Note You should select which interfaces to configure so that traffic is encrypted at the outbound interface of the local peer router, and traffic is decrypted at the input interface of the remote peer.

Setting Up Encryption Access List

Encryption access lists are used in this step to define which IP packets will be encrypted and which IP packets will not be encrypted. Encryption access lists are defined using extended IP access lists. (Normally, IP access lists are used to filter traffic. Encryption access lists are *not* used to filter traffic but are used to specify which packets to encrypt or not encrypt.)

Set up encryption access lists for IP packet encryption by using either of the following commands in global configuration mode:

Command	Purpose
access-list *access-list-number* [**dynamic** dynamic-name [**timeout** *minutes*]] {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log**] or **ip access-list extended** *name* Follow with **permit** and **deny** statements as appropriate.	Specifies conditions to determine which IP packets will be encrypted. (Enables or disables encryption for traffic that matches these conditions.)[1]

[1] You specify conditions using an IP access list designated by either a number or a name. The **access-list** command designates a numbered access list; the **ip access-list extended** command designates a named access list.

Using the **permit** keyword will cause the selected traffic that is passed between the specified source and destination addresses to be encrypted/decrypted by peer routers. Using the **deny** keyword prevents that traffic from being encrypted/decrypted by peer routers.

The encryption access list you define at the local router must have a "mirror-image" encryption access list defined at the remote router, so that traffic that is encrypted locally is decrypted at the remote peer.

See the *Cisco IOS Security Command Reference* for complete details about the extended IP access list commands.

The encryption access list you define will be applied to an interface as an outbound encryption access list after you define a crypto map and apply the crypto map to the interface. (These two tasks are described in the next sections.)

Caution When you create encryption access lists, Cisco recommends *against* using the **any** keyword to specify source or destination addresses. Using the **any** keyword could cause extreme problems if a packet enters your router and is destined for a router that is not configured for encryption. This would cause your router to attempt to set up an encryption session with a nonencrypting router.

Note If your encryption access lists define more than 100 distinct source addresses or more than 10 destination addresses for a given source address, you need to change certain defaults as described later in the section "Changing Encryption Access List Limits."

Note If you view your router's access lists by using a command such as **show ip access-lists,** *all* extended IP access lists will be shown in the command output. This includes extended IP access lists that are used for traffic filtering purposes, as well as those that are used for encryption. The **show** command output does not differentiate between the two uses of the extended access lists.

About Crypto Maps

Crypto maps are used to specify which DES encryption algorithm(s) will be used in conjunction with each access list defined in the previous step. Crypto maps are also used to identify which peer routers will provide the remote end encryption services.

Crypto map entries with the same crypto map name (but different map sequence numbers) are grouped into a crypto map set. Later, you will apply these crypto map sets to interfaces; then, all IP traffic passing through the interface is evaluated against the applied crypto map set. If a crypto map entry sees outbound IP traffic that should be protected and the crypto map specifies the use of IKE, a security association is negotiated with the remote peer according to the parameters included in the crypto map entry; otherwise, if the crypto map entry specifies the use of manual security associations, a security association should have already been established via configuration. (If a dynamic crypto map entry sees outbound traffic that should be protected and no security association exists, the packet is dropped.)

The policy described in the crypto map entries is used during the negotiation of security associations. If the local router initiates the negotiation, it will use the policy specified in the static crypto map entries to create the offer to be sent to the specified IPSec peer. If the IPSec peer initiates the negotiation, the local router will check the policy from the static crypto map entries, as well as any referenced dynamic crypto map entries, to decide whether to accept or reject the peer's request (offer).

If you create more than one crypto map entry for a given interface, use the *seq-num* of each map entry to rank the map entries: the lower the *seq-num*, the higher the priority. At the interface that has the crypto map set, traffic is evaluated against higher priority map entries first.

Defining Crypto Maps

You must define exactly one crypto map for each interface that will send encrypted data to a peer encrypting router. You can apply only one crypto map set to a single interface. Multiple interfaces can share the same crypto map set if you want to apply the same policy to multiple interfaces.

To define a crypto map, use the following commands. The first command is used in global configuration mode; the other commands are used in crypto map configuration mode.

Step	Command	Purpose
1	`crypto map` *`map-name seq-num`* `[cisco]`	Names the crypto map. (Executing this command causes you to enter the crypto map configuration mode.)
2	`set peer` *`key-name`*	Specifies the remote peer router.

Step	Command	Purpose
3	`match address [access-list-id | name]`	Specifies at least one encryption access list.

4	`set algorithm des [cfb-8 | cfb-64]`
	or
	`set algorithm 40-bit-des [cfb-8 | cfb-64]`
		Specifies at least one DES encryption algorithm. (This must be an algorithm you previously enabled.)

Note If you are running an exportable image, you can only specify 40-bit variations of DES. You cannot enable or use the basic DES algorithms, which are not available with exportable images.

To define an additional, different set of parameters for the same interface, repeat the steps in the previous task list, using the same *map-name* but use a different *seq-num* for the crypto map command. For more information about this, refer to the **crypto map** command description in the "Cisco Encryption Technology Commands" chapter of the *Cisco IOS Security Command Reference.*

Applying Crypto Maps to Interfaces

This step puts into effect the crypto maps just defined. You must apply exactly one crypto map set to each interface (physical or logical) that will encrypt outbound data and decrypt inbound data. This interface provides the encrypted connection to a peer encrypting router. An interface will not encrypt/decrypt data until you apply a crypto map to the interface.

To apply a crypto map to an interface, use the following command in interface configuration mode:

Command	Purpose
`crypto map map-name`	Applies a crypto map to an interface.

Backing Up Your Configuration

Cisco recommends that after you configure your router for encryption, you make a backup of your configuration. (Be careful to restrict unauthorized access of this backed-up configuration.)

You can learn how to back up your configuration in the "Modifying, Downloading, and Maintaining Configuration Files" chapter of the *Cisco IOS Configuration Fundamentals Configuration Guide.*

Configuring Encryption with GRE Tunnels

When GRE tunnel endpoints are located at the peer encrypting routers, you can configure encryption so that all traffic through the GRE tunnel is encrypted.

Note that you cannot selectively encrypt GRE tunnel traffic: either all the GRE tunnel traffic is encrypted, or no GRE tunnel traffic is encrypted.

To configure encryption with GRE tunnels, perform the same basic tasks described previously in the section "Configuring Encryption." However, you also must follow the additional instructions described next (for two cases):

- Encrypting Only GRE Tunnel Traffic
- Encrypting GRE Tunnel Traffic and Other Traffic

For examples of configuring encryption with a GRE tunnel, see the section "Configuring Encryption with GRE Tunnels Examples" later in this chapter.

Encrypting Only GRE Tunnel Traffic

To encrypt only traffic through the GRE tunnel, follow these two additional instructions:

- When you set up your encryption access list, the list should contain only one criteria statement. In this one statement, specify **gre** as the protocol, specify the tunnel source address as the source, and specify the tunnel destination address as the destination.
- Apply the crypto map to both the physical interface and to the tunnel interface. (Without GRE tunnels, you only had to apply the crypto map to the physical interface.)

  Remember to apply a crypto map to the physical interface and tunnel interface at both ends of the GRE tunnel.

Encrypting GRE Tunnel Traffic and Other Traffic

To encrypt both GRE tunnel traffic and other specified non-GRE tunnel traffic, follow these three additional instructions:

- Create two separate encryption access lists as follows:
  - The first encryption access list should contain only one criteria statement. In this one statement, specify **gre** as the protocol, specify the tunnel source address as the source, and specify the tunnel destination address as the destination.
  - In the second encryption access list, specify which non-GRE traffic should be encrypted. (For example, you could specify **tcp** as a protocol, and specify a subnet source/wildcard and a subnet destination/wildcard.)
- Create two separate crypto map sets as follows:
  - In the first crypto map set, specify a single crypto map that includes the first encryption access list, along with a DES algorithm and the remote peer.
  - In the second crypto map set, include at least two crypto map subdefinitions. The first subdefinition should exactly match the statements in the first crypto map. The second subdefinition should specify the second encryption access list, a DES algorithm, and the remote peer.
- Apply the first crypto map set to the tunnel interface, and apply the second crypto map set to the physical interface. (Without GRE tunnels, you only have to apply one crypto map to the physical interface.)

  Remember to apply a crypto map set to the physical interface and tunnel interface at both ends of the GRE tunnel.

Configuring Encryption with an ESA in a VIP2

To configure encryption with an Encryption Service Adaptor (ESA), there are additional instructions that you must follow, in addition to the basic encryption configuration tasks described previously in the section "Configuring Encryption."

This section describes configuration for an ESA plugged into a VIP2 on a Cisco RSP7000 or 7500 series router.

To configure encryption with an ESA plugged into a VIP2, complete these tasks in this order:

1. Resetting the ESA
2. Performing Additional Encryption Configuration

Note In Cisco 7500 and RSP7000-equipped Cisco 7000 systems, the ESA requires a VIP2-40 for operation and it must be installed in PA slot 1. If you ever remove and reinstall the ESA or the VIP2, you must reset the ESA again.

For examples of ESA-specific configuration tasks, see the section "ESA-Specific Encryption Configuration Task Examples" later in this chapter.

Resetting the ESA

If the ESA has never been used before, or if it has been removed and reinstalled, the ESA's "Tampered" LED is lit and it must be reset.

If you do not reset the ESA in a VIP2, the ESA crypto engine will not be used; instead, the VIP2 crypto engine will govern the adjoining VIP2 port interfaces (and the Cisco IOS crypto engine will govern the other router interfaces).

To reset an ESA, complete one of the following bulleted tasks:

- Reset an ESA that has never been used before (or was previously used and you know the ESA password), by using the following commands in global configuration mode:

Step	Command	Purpose
1	`crypto card clear-latch` *`slot`*	Resets the ESA by clearing the ESA hardware latch.
2	`password`	When prompted, creates a new password for the ESA or type the ESA password previously assigned.

- Reset an ESA that was previously used, but you do not know the ESA password, by using the following command in global configuration mode:

Command	Purpose
`crypto key zeroize dss` *`slot`*	Clears the ESA. (This deletes all DSS keys for the ESA.)

Performing Additional Encryption Configuration

If the router, VIP2, and ESA were all previously configured for encryption, you might not need to complete any additional configuration. However, you will need additional configuration in at least the following cases (see the section "Configuring Encryption" for descriptions of the tasks):

- If you have any concern that the old ESA keys are compromised, you should regenerate and exchange new DSS keys for the ESA. (Use the same ESA *key-name* previously assigned.)
- If the ESA was relocated and now governs different interfaces than before, either all peer routers must update their crypto maps to reflect the changed peers, or you must regenerate and exchange new DSS keys for the ESA, assigning the *key-name* that is currently in the peer routers' crypto maps.
- If you previously reset the ESA with the **crypto key zeroize dss** command because you did not know the ESA password, you must at a minimum generate and exchange DSS keys for the ESA crypto engine.

As always, remember to back up your configuration when you are done.

Configuring Encryption with an ESA in a Cisco 7200 Series Router

To configure encryption with an Encryption Service Adaptor (ESA), there are additional instructions that you must follow in addition to the basic encryption configuration tasks described previously in the section "Configuring Encryption."

This section describes configuration for an ESA plugged into a Cisco 7200 series router.

For examples of ESA-specific configuration tasks, see the "ESA-Specific Encryption Configuration Task Examples" section later in this chapter.

Required Tasks

Complete the following tasks in this order (see following sections for descriptions):

1. Resetting the ESA
2. Performing Additional Encryption Configuration
3. Enabling the ESA

Note If you ever remove and reinstall the ESA, you must reset the ESA again and re-enable the ESA.

Optional Tasks

You can optionally complete these additional tasks (see following sections for descriptions):

- Selecting a Crypto Engine (After encryption is configured, you might want to change which crypto engine to use—the Cisco IOS crypto engine or the ESA crypto engine.)
- Deleting DSS Keys (If you ever remove or relocate the ESA or the Cisco 7200, you might want to delete DSS keys, to reduce any potential security risk.)

Resetting the ESA

If the ESA has never been used before, or if it has been removed and reinstalled, the ESA's "Tampered" LED is lit and it must be reset.

To reset an ESA in a Cisco 7200 series router, complete one of the following bulleted tasks:

- Reset an ESA that has never been used before by using the following commands in global configuration mode:

Step	Command	Purpose
1	`crypto card clear-latch` *`slot`*	Resets the ESA by clearing the ESA hardware latch.
2	`password`	When prompted, creates a new password for the ESA.

- Reset a previously used ESA that needs additional configuration (for example, the ESA's previous configuration was not complete or is uncertain; or you know you want to generate new DSS keys for the ESA; or the router is not configured for encryption) by using the following commands in global configuration mode:

Step	Command	Purpose
1	`crypto card clear-latch` *slot*	Resets the ESA by clearing the ESA hardware latch.
2	`password`	When prompted, type the ESA password previously assigned.
3	`no`	If prompted to enable the ESA, type **no**.

- Reset a previously used ESA when encryption configuration is already complete and you are ready to start encrypting traffic using the ESA crypto engine by using the following commands in global configuration mode:

Step	Command	Purpose
1	`crypto card clear-latch` *slot*	Resets the ESA by clearing the ESA hardware latch.
2	`password`	When prompted, type the ESA password previously assigned.
3	`yes`	When prompted to enable the ESA, type **yes**.

Note After you reset the ESA as just described, the ESA will automatically become active and begin encrypting traffic. For this case only, you do not need to complete any additional encryption configuration. (But as always, be sure to back up your configuration.)

- Reset a previously used ESA when you do not know the ESA password by using the following command in global configuration mode:

Command	Purpose
`crypto key zeroize dss` *slot*	Clears the ESA. (This deletes all DSS keys for the ESA.)

Performing Additional Encryption Configuration

After you reset the ESA in a Cisco 7200 series router, continue configuring encryption by following the instructions in one of the following bullets:

- If the router and ESA were never previously configured for encryption, complete all the tasks described earlier in the section "Configuring Encryption," then enable the ESA as described in the next section, "Enabling the ESA."
- If the ESA was never previously configured for encryption, but the router is configured for encryption, complete only the following two tasks described earlier in the section "Configuring Encryption":
  - — Generating DSS Public/Private Keys (for the ESA crypto engine)
  - — Exchanging DSS Public Keys (for the ESA crypto engine)

  After you generate and exchange DSS keys for the ESA crypto engine, enable the ESA as described in the next section, "Enabling the ESA."

- If the router and ESA are both already configured for encryption, you might only need to enable the ESA as described in the next section, "Enabling the ESA." However, in at least the following cases you will need additional configuration before you enable the ESA (see the section "Configuring Encryption" for descriptions of the tasks):
    - If the ESA has DSS keys generated but not exchanged with the peer routers, you must exchange the keys.
    - If you have any concern that the ESA's DSS keys are compromised, you should regenerate and exchange new DSS keys for the ESA, using the same *key-name* assigned to the router DSS keys.
    - If the ESA was relocated from a different router, regenerate and exchange DSS keys, using the same *key-name* assigned to the router DSS keys.
    - If you previously reset the ESA with the **crypto key zeroize dss** command because you did not know the ESA password, you must at a minimum generate and exchange DSS keys for the ESA crypto engine.

Enabling the ESA

Enable an ESA in a Cisco 7200 series router by using the following command in global configuration mode:

Command	Purpose
`crypto card enable` *`slot`*	Enables the ESA.

Note If the Cisco IOS crypto engine is currently encrypting traffic when you enable the ESA, the session will be torn down, and a new session will be established using the ESA crypto engine. This could cause a momentary delay for encrypted traffic.

As always, remember to back up your configuration when you are done.

Selecting a Crypto Engine

This is an optional task.

After encryption is configured on a Cisco 7200 series router with an ESA, you might want to change which crypto engine to use—the Cisco IOS crypto engine or the ESA crypto engine. This section describes how to switch from one crypto engine to the other.

You should only select a crypto engine if the engine is fully configured for encryption.

If you boot the router with an operational ESA installed, the ESA will be the active crypto engine upon bootup, by default. Otherwise, the Cisco IOS crypto engine will be the default active crypto engine.

Note If any encryption session is in progress when you switch from one crypto engine to the other, the session will be torn down, and a new session will be established using the newly selected crypto engine. This could cause a momentary delay for encrypted traffic.

Selecting the Cisco IOS Crypto Engine

If the ESA crypto engine is encrypting traffic, but you want to cause the Cisco IOS crypto engine to encrypt the traffic instead, you can switch to the Cisco IOS crypto engine without removing the ESA. (You might want to do this for testing purposes.)

Caution Before you switch to the Cisco IOS crypto engine, be sure that the Cisco IOS crypto engine is configured with DSS keys generated and exchanged; otherwise, you will lose encryption capability when you switch engines.

Select the Cisco IOS crypto engine by using the following command in global configuration mode:

Command	Purpose
`crypto card shutdown` *slot*	Shuts down the ESA.

After you select the Cisco IOS crypto engine, the Cisco IOS crypto engine will be the active engine, governing the router interfaces. The Cisco IOS crypto engine will perform the encryption services, and the ESA will be inactive.

Selecting the ESA Crypto Engine

If the Cisco IOS crypto engine is encrypting traffic, but you want to cause an installed ESA crypto engine to encrypt the traffic instead, you can switch to the ESA crypto engine.

Caution Before you switch to the ESA crypto engine, be sure that the ESA crypto engine is configured with DSS keys generated and exchanged; otherwise, you will lose encryption capability when you switch engines.

Select the ESA crypto engine by using the following command in global configuration mode:

Command	Purpose
`crypto card enable` *slot*	Enables the ESA.

After you select the ESA crypto engine, the ESA crypto engine will be the active engine, governing the router interfaces. The ESA crypto engine will perform encryption services for the router, and the Cisco IOS crypto engine will be inactive.

Deleting DSS Keys

This is an optional task.

If you ever remove or relocate the ESA or the Cisco 7200, or if the DSS keys ever become compromised, or if you want to turn encryption off at the router, you might want to delete DSS keys to reduce any potential security risk. This section describes how to delete a DSS key pair for an ESA or for a Cisco 7200 series router.

To delete DSS keys, use the following commands beginning in EXEC mode:

Step	Command	Purpose
1	`show crypto key mypubkey dss`	Displays all existing sets of DSS keys (ESA and Cisco IOS keys).
2	`show crypto engine configuration`	Determines the current (active) crypto engine.

Step	Command	Purpose
3	`crypto card enable` *slot* (switch to the Cisco IOS crypto engine) or `crypto card shutdown` *slot* (switch to the ESA crypto engine)	If the current engine is not the engine for which you want to delete keys, change engines. (When you delete keys, the software deletes keys for the current active engine.)
4	`show crypto engine configuration`	Verifies that the current crypto engine is the engine for which you want to delete keys.
5	`crypto key zeroize dss` (for the Cisco IOS crypto engine) or `crypto key zeroize dss` *slot* (for the ESA crypto engine)	Deletes the DSS keys for the current crypto engine.

After you delete DSS keys for a crypto engine, if you ever want to use that engine for encryption, you must regenerate and exchange new DSS keys for that engine. For the ESA crypto engine, you must also enable the ESA.

Customizing Encryption (Configuring Options)

This section describes options that you can configure to customize encryption on a router:

- Defining Time Duration of Encrypted Sessions
- Shortening Session Setup Times by Pregenerating DH Numbers
- Changing Encryption Access List Limits

Defining Time Duration of Encrypted Sessions

The default time duration of an encrypted session is 30 minutes. After the default time duration expires, an encrypted session must be renegotiated if encrypted communication is to continue. You can change this default to extend or shorten the time of encrypted sessions.

You might want to shorten session times if you believe that there is a risk of compromised session keys.

You might want to extend session times if your system has trouble tolerating the interruptions caused when sessions are renegotiated.

Change the time duration of encrypted sessions by using at least the first of the following commands in global configuration mode:

Step	Command	Purpose
1	`crypto cisco key-timeout` *minutes*	Defines maximum time duration of encrypted sessions.
2	`show crypto cisco key-timeout`	Displays defined time duration of encrypted sessions.

Shortening Session Setup Times by Pregenerating DH Numbers

Diffie-Hellman (DH) numbers are generated in pairs during the setup of each encrypted session. (DH numbers are used during encrypted session setup to compute the DES session key.) Generating these numbers is a CPU-intensive activity, which can make session setup slow—especially for low-end routers. To speed up session setup time, you may choose to pregenerate DH numbers. It is usually necessary to pregenerate only one or two DH numbers.

Pregenerate DH numbers by using the following command in global configuration mode:

Command	Purpose
crypto cisco pregen-dh-pairs *count* [*slot*]	Pregenerates DH numbers.

Changing Encryption Access List Limits

When you configure encryption access lists, you configure source and destination pairs in criteria statements. Any traffic that matches the criteria is then encrypted.

By default, the maximum number of distinct sources (host or subnets) that you can define in an encryption access list is 100. Also, the maximum number of distinct destinations that you can define for any given source address is 10. For example, if you define six different source addresses, you can define up to 10 destination addresses for each of the six sources, for a total of 60 access list criteria statements.

Why Do These Limits Exist?

These limits exist because of the amount of memory that must be reserved for encryption connections. If there are more potential connections, there must be more memory preallocated.

When Should the Limits Be Changed?

For most situations, the defaults of 100 maximum sources and 10 maximum destinations per source are sufficient. Cisco recommends that you do not change the defaults unless you actually exceed the number of sources or destinations per source.

However, in some situations you might want to change one or both of these maximum values. For example, if more than 10 remote sites need to connect to one server behind your router, then you need more than 10 destination addresses (one for each remote site) to pair up with the server's source address in the local router's encryption access list. In this case, you need to change the default of 10 maximum destination addresses per source address.

When changing limits, you should consider the amount of memory that will be allocated. In general, if you increase one value, decrease the other value. This prevents your router from running out of memory because too much memory was preallocated.

How Much Memory Is Preallocated If the Limits Are Changed?

The amount of memory reserved for encrypted connections changes if you change the defaults.

For every additional source, the following additional bytes of memory will be allocated:

```
64 + (86 x the specified number of maximum destinations)
```

For every additional destination, the following additional bytes of memory will be allocated:

```
68 x the specified number of maximum sources
```

For example, if you specify 5 maximum sources, and 250 maximum destinations per source, the memory allocated for encryption connections is calculated as follows:

```
{5 x [64 + (68 x 250)]} + {250 x (68 x 5)} = 170320 bytes
```

How Are the Limits Changed?

Change the default limits by using one or both of the following commands in global configuration mode, then reboot the router for the changes to take effect:

Command	Purpose
`crypto cisco entities` *number*	Changes the maximum number of distinct sources (hosts or subnets) that you can define in the encryption access list statements.
`crypto cisco connections number`	Changes the maximum number of destinations (hosts or subnets) per source that you can define in the encryption access list statements.

Note You must reboot the router for these changes to take effect.

For an example of changing these values, see the section "Changing Encryption Access List Limits Example" later in this chapter.

Turning Off Encryption

You can turn off encryption for certain router interfaces, or you can turn off encryption completely for the entire router.

- To turn off encryption at all the interfaces governed by a single crypto engine, you can delete DSS keys for that engine. Deleting DSS keys is described in this section.
- To turn off encryption at certain random interfaces, you can remove the crypto maps from the interfaces with the **no crypto map (interface configuration)** command.
- To turn off encryption completely for a router, you can delete the DSS keys for all the router's crypto engines. Deleting DSS keys is described in this section.

Deleting DSS keys deconfigures encryption for the crypto engine and also reduces security risk by ensuring that the keys cannot be misused if you lose physical control over the router or ESA.

After you delete DSS keys for a crypto engine, you will not be able to perform encryption on the interfaces governed by that crypto engine.

Caution DSS keys cannot be recovered after they have been deleted. Use this function only after careful consideration.

For all platforms other than Cisco 7200 series routers, to delete DSS public/private keys for a crypto engine, use the following command in global configuration mode:

Command	Purpose
`crypto key zeroize dss` [*slot*][1]	Deletes DSS keys for a crypto engine.

[1] Only Cisco 7200 and 7500 series routers require the *slot* argument.

For a Cisco 7200 series router, to delete DSS public/private keys for a crypto engine, refer to the section "Deleting DSS Keys" earlier in this chapter.

Testing and Troubleshooting Encryption

This section discusses how to verify your configuration and the correct operation of encryption. This section also discusses diagnosing encryption problems.

You should complete all the required configuration tasks (as described earlier in this chapter) before trying to test or troubleshoot your encryption configuration.

This section includes the following topics:

- Testing the Encryption Configuration
- Diagnosing Connection Problems
- Diagnosing Other Miscellaneous Problems
- Using Debug Commands

Testing the Encryption Configuration

If you want to test the encryption setup between peer routers, you can attempt to manually establish a session using the IP address of a local host and a remote host which have been specified in an encryption access list. (The encryption list must be specified in a crypto map definition, and that crypto map must be applied to an interface before this test will be successful.)

To test the encryption setup, use the following commands in privileged EXEC mode:

Step	Command	Purpose
1	`test crypto initiate-session` *src-ip-addr dst-ip-addr map-name seq-num*	Sets up a test encryption session.
2	`show crypto cisco connections`	Displays the connection status.

An example at the end of this chapter explains how to interpret the **show crypto cisco connections** command output.

Diagnosing Connection Problems

If you need to verify the state of a connection, you can use the following commands in privileged EXEC mode:

Command	Purpose
`show crypto cisco connections`	Checks status of all encryption connections.
`show crypto map`	Checks status of a crypto map.
`show crypto engine connections active`	Checks that connection is established and that packets are being encrypted.

Diagnosing Other Miscellaneous Problems

When using encryption, you might encounter some of these problems:

- Dropped Packets
- Difficulty Establishing Telnet Sessions
- Invalid DSS Public/Private Keys
- ESA Crypto Engine Not Active
- Password Requested When You Generate DSS Keys
- Router Hanging

Dropped Packets

Packets are normally dropped while an encrypted session is being set up. If this poses a problem for your network, you should extend the length of encryption sessions as described previously in the section "Defining Time Duration of Encrypted Sessions." The longer the session time, the fewer the interruptions caused by session renegotiation.

Packets might also be dropped if you switch crypto engines in a Cisco 7200 series router with an ESA. If this is a problem, you should only switch crypto engines when encrypted traffic is light.

IP fragments are always dropped on VIP2 interfaces, because IP fragmentation is not supported with encryption on VIP2 interfaces.

Difficulty Establishing Telnet Sessions

Hosts might experience difficulty in establishing Telnet sessions if the session uses two encrypting peer routers to create the connection. This difficulty is more likely to occur if the peer routers are low-end routers such as Cisco 2500 series routers. Telnet sessions can fail to be established when a Telnet connection attempt times out before the encrypted session setup is complete.

If a Telnet session fails to establish, the host should wait a short time (a few seconds might be sufficient), and then attempt the Telnet connection again. After the short wait, the encrypted session setup should be complete, and the Telnet session can be established. Enabling pregeneration of DH numbers (described later in this chapter) might also help by speeding up encryption session connection setup times.

Invalid DSS Public/Private Keys

If NVRAM fails, or if your ESA is tampered with or replaced, DSS public/private keys will no longer be valid. If this happens, you will need to regenerate and re-exchange DSS keys. Generating and exchanging DSS keys are described earlier in the section "Configuring Encryption."

ESA Crypto Engine Not Active

If an installed ESA is not active when you boot a router, the router displays a message similar to this message, indicating that the router switched over to the Cisco IOS crypto engine:

```
There are no keys on the ESA in slot 2- ESA not enabled

...switching to SW crypto engine
```

You can also determine if the ESA crypto engine is not active by using the **show crypto engine brief** command—look at the "crypto engine state" field in the output. If no crypto engine is active, the state field indicates "pending."

The ESA crypto engine will not be active if you removed and reinstalled the ESA, if the ESA was tampered with, or if encryption is not configured correctly for the ESA.

If the Cisco IOS crypto engine is active, but you want to use the ESA crypto engine instead, make sure that the ESA crypto engine is reset (**crypto card clear-latch** command), and for Cisco 7200 series routers, also make sure that the ESA crypto engine is enabled (**crypto card enable** command). You might also need to complete or verify additional configuration; refer to the instructions for configuring encryption with an ESA in the earlier sections "Configuring Encryption with an ESA in a VIP2" or "Configuring Encryption with an ESA in a Cisco 7200 Series Router."

To verify that the ESA has DSS keys, you can use the **show crypto card** command and look at the "DSS Key set" field in the output. If the field contains "Yes," the ESA has DSS keys generated and stored. In this case, you might only need to reset and enable the ESA to make it active.

Password Requested When You Generate DSS Keys

If you attempt to generate DSS keys for the Cisco IOS crypto engine on a Cisco 7200 series router with an installed ESA without DSS keys, the router might assume that you are trying to generate keys for the ESA and prompt for the ESA password.

- If you want to generate keys for the ESA, you must supply the ESA password. If you do not know the password, you must reset the ESA as described in the section "Configuring Encryption with an ESA in a Cisco 7200 Series Router" earlier in this chapter.
- If you want to generate keys for the Cisco IOS crypto engine, not the ESA crypto engine, you must select the Cisco IOS crypto engine to make it the active engine.

  Select the Cisco IOS crypto engine by using the following command in global configuration mode:

Command	Purpose
`crypto card shutdown` *`slot`*	Shuts down the ESA.

- When the Cisco IOS crypto engine is active, you can generate keys for the router, and you will not be prompted for a password.

Router Hanging

If you remove a configured ESA from a VIP2, you must reboot the router. If you do not, the router might hang when it tries to access the absent ESA.

Using Debug Commands

Debug commands are also available to assist in problem solving. These commands are documented in the *Cisco IOS Debug Command Reference*.

Encryption Configuration Examples

The following sections provide examples of configuring and testing your router for CET:

- Generating DSS Public/Private Keys Example
- Exchanging DSS Public Keys Example
- Enabling DES Encryption Algorithms Example
- Setting Up Encryption Access Lists, Defining Crypto Maps, and Applying Crypto Maps to Interfaces Examples
- Changing Encryption Access List Limits Example
- Configuring Encryption with GRE Tunnels Examples
- ESA-Specific Encryption Configuration Task Examples
- Deleting DSS Keys Example
- Testing the Encryption Connection Example

Generating DSS Public/Private Keys Example

The following example illustrates two encrypting peer routers (named Apricot and Banana) generating their respective DSS public/private keys. Apricot is a Cisco 2500 series router. Banana is a Cisco 7500 series router with an RSP in chassis slot 4 and an ESA/VIP2 in chassis slot 2.

Apricot

```
Apricot(config)# crypto key generate dss Apricot
Generating DSS keys .... [OK]
Apricot(config)#
```

Banana

```
Banana(config)# crypto key generate dss BananaIOS 4
Generating DSS keys .... [OK]
Banana(config)# crypto key generate dss BananaESA 2
% Initialize the crypto card password. You will need
   this password in order to generate new signature
   keys or clear the crypto card extraction latch.

Password: <passwd>

Re-enter password: <passwd>

Generating DSS keys .... [OK]
Banana(config)#
```

The password entered in this example is a new password that you create when you generate DSS keys for an ESA crypto engine for the first time. If you ever generate DSS keys a second time for the same ESA crypto engine, you must use the same password to complete the key regeneration.

Exchanging DSS Public Keys Example

The following is an example of a DSS public key exchange between two peer encrypting routers (Apricot and Banana). Apricot is a Cisco 2500 series router, and Banana is a Cisco 7500 series router with an ESA. In this example, Apricot sends its Cisco IOS DSS public key, and Banana sends its ESA DSS public key. DSS keys have already been generated as shown in the previous example.

Before any commands are entered, one administrator must call the other administrator. After the phone call is established, the two administrators decide which router is "PASSIVE" and which is "ACTIVE" (an arbitrary choice). In this example, router Apricot is ACTIVE and router Banana is PASSIVE. To start, PASSIVE enables a connection as follows:

Banana (PASSIVE)

```
Banana(config)# crypto key exchange dss passive
Enter escape character to abort if connection does not complete.
Wait for connection from peer[confirm]<Return>
Waiting ....
```

PASSIVE must wait while ACTIVE initiates the connection and sends a DSS public key.

Apricot (ACTIVE)

```
Apricot(config)# crypto key exchange dss 192.168.114.68 Apricot
Public key for Apricot:
   Serial Number 01461300
   Fingerprint   0F1D 373F 2FC1 872C D5D7

Wait for peer to send a key[confirm]<Return>
Waiting ....
```

After ACTIVE sends a DSS public key, the key's serial number and Fingerprint display on both terminals, as shown previously and as follows:

Banana (PASSIVE)

```
Public key for Apricot:
   Serial Number 01461300
   Fingerprint   0F1D 373F 2FC1 872C D5D7
Add this public key to the configuration? [yes/no]: y
```

Now the two administrators both must verbally verify that their two screens show the same serial number and Fingerprint. If they do, PASSIVE will accept the DSS key as shown previously by typing **y**, and continue by sending ACTIVE a DSS public key:

```
Send peer a key in return[confirm]<Return>
Which one?

BananaIOS? [yes]: n
BananaESA? [yes]: <Return>
Public key for BananaESA:
   Serial Number 01579312
   Fingerprint   BF1F 9EAC B17E F2A1 BA77

Banana(config)#
```

Both administrators observe Banana's serial number and Fingerprint on their screens. Again, they verbally verify that the two screens show the same numbers.

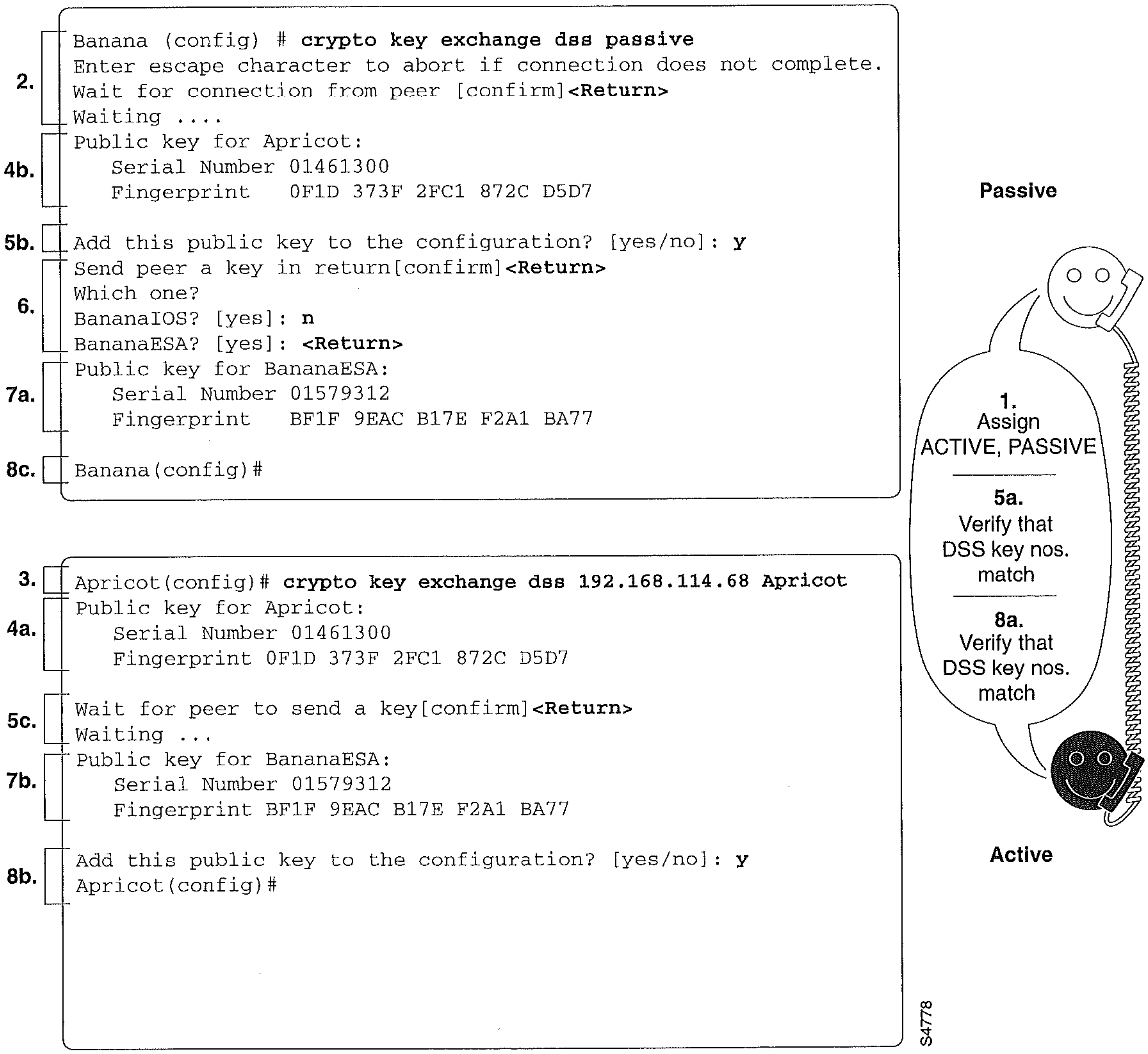
Apricot (ACTIVE):

```
Public key for BananaESA:
   Serial Number 01579312
   Fingerprint   BF1F 9EAC B17E F2A1 BA77

Add this public key to the configuration? [yes/no]: y
Apricot(config)#
```

ACTIVE accepts Apricot's DSS public key. Both administrators hang up the phone and the key exchange is complete.

Figure 30 shows complete screens of the two routers. The steps are numbered on the figure to show the sequence of the entire exchange.

Figure 30 DSS Public Key Exchange (Numbers Indicate Sequence of Events)



```
2.   Banana (config) # crypto key exchange dss passive
     Enter escape character to abort if connection does not complete.
     Wait for connection from peer [confirm]<Return>
     Waiting ....
4b.  Public key for Apricot:
        Serial Number 01461300
        Fingerprint   0F1D 373F 2FC1 872C D5D7

5b.  Add this public key to the configuration? [yes/no]: y
6.   Send peer a key in return[confirm]<Return>
     Which one?
     BananaIOS? [yes]: n
     BananaESA? [yes]: <Return>
7a.  Public key for BananaESA:
        Serial Number 01579312
        Fingerprint   BF1F 9EAC B17E F2A1 BA77

8c.  Banana(config)#
```

```
3.   Apricot(config)# crypto key exchange dss 192.168.114.68 Apricot
4a.  Public key for Apricot:
        Serial Number 01461300
        Fingerprint 0F1D 373F 2FC1 872C D5D7

5c.  Wait for peer to send a key[confirm]<Return>
     Waiting ...
7b.  Public key for BananaESA:
        Serial Number 01579312
        Fingerprint BF1F 9EAC B17E F2A1 BA77

8b.  Add this public key to the configuration? [yes/no]: y
     Apricot(config)#
```

Passive
1. Assign ACTIVE, PASSIVE
5a. Verify that DSS key nos. match
8a. Verify that DSS key nos. match
Active
S4778

Enabling DES Encryption Algorithms Example

In this example, a router (Apricot) globally enables two DES algorithms: the basic DES algorithm with 8-bit Cipher Feedback (CFB), and the 40-bit DES algorithm with 8-bit CFB. Another router (Banana) globally enables three DES algorithms: the basic DES algorithm with 8-bit CFB, the basic DES algorithm with 64-bit CFB, and the 40-bit DES algorithm with 8-bit CFB.

The following commands are entered in global configuration mode.

Apricot

```
crypto cisco algorithm des cfb-8
crypto cisco algorithm 40-bit-des cfb-8
```

Banana

```
crypto cisco algorithm des cfb-8
crypto cisco algorithm des cfb-64
crypto cisco algorithm 40-bit-des cfb-8
```

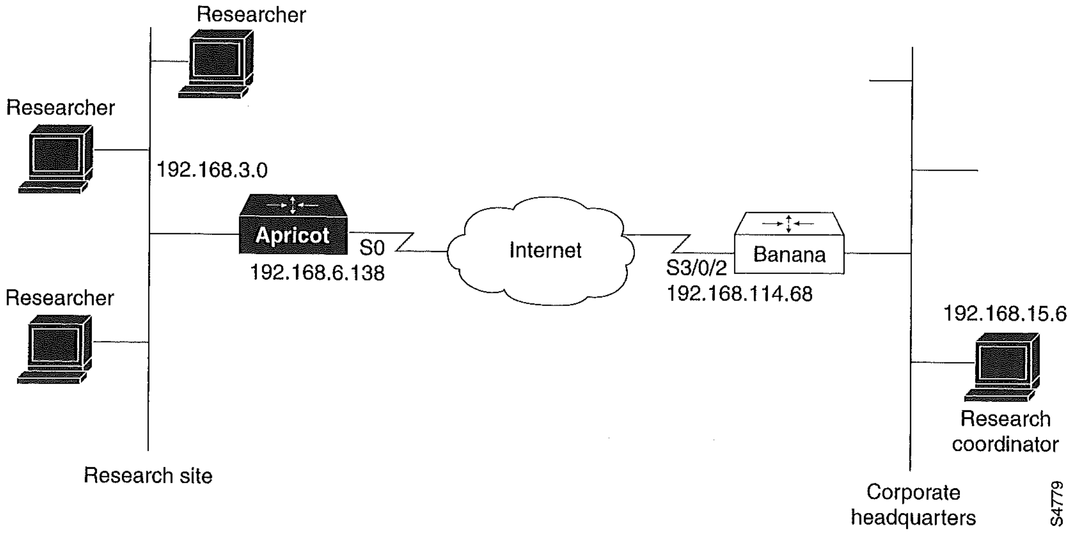
Setting Up Encryption Access Lists, Defining Crypto Maps, and Applying Crypto Maps to Interfaces Examples

The following two examples show how to set up interfaces for encrypted transmission. Participating routers will be configured as encrypting peers for IP packet encryption.

Example 1

In the first example, a team of researchers at a remote site communicate with a research coordinator at headquarters. Company-confidential information is exchanged by IP traffic that consists only of TCP data. Figure 31 shows the network topology.

Figure 31 Example 1 Network Topology



Apricot is a Cisco 2500 series router, and Banana is a Cisco 7500 series router with an ESA/VIP2 in chassis slot 3.

Apricot

```
Apricot(config)# access-list 101 permit tcp 192.168.3.0 0.0.0.15 host 192.168.15.6
Apricot(config)# crypto map Research 10
Apricot(config-crypto-map)# set peer BananaESA
Apricot(config-crypto-map)# set algorithm des cfb-8
Apricot(config-crypto-map)# match address 101
Apricot(config-crypto-map)# exit
Apricot(config)# interface s0
Apricot(config-if)# crypto map Research
Apricot(config-if)# exit
Apricot(config)#
```

Banana

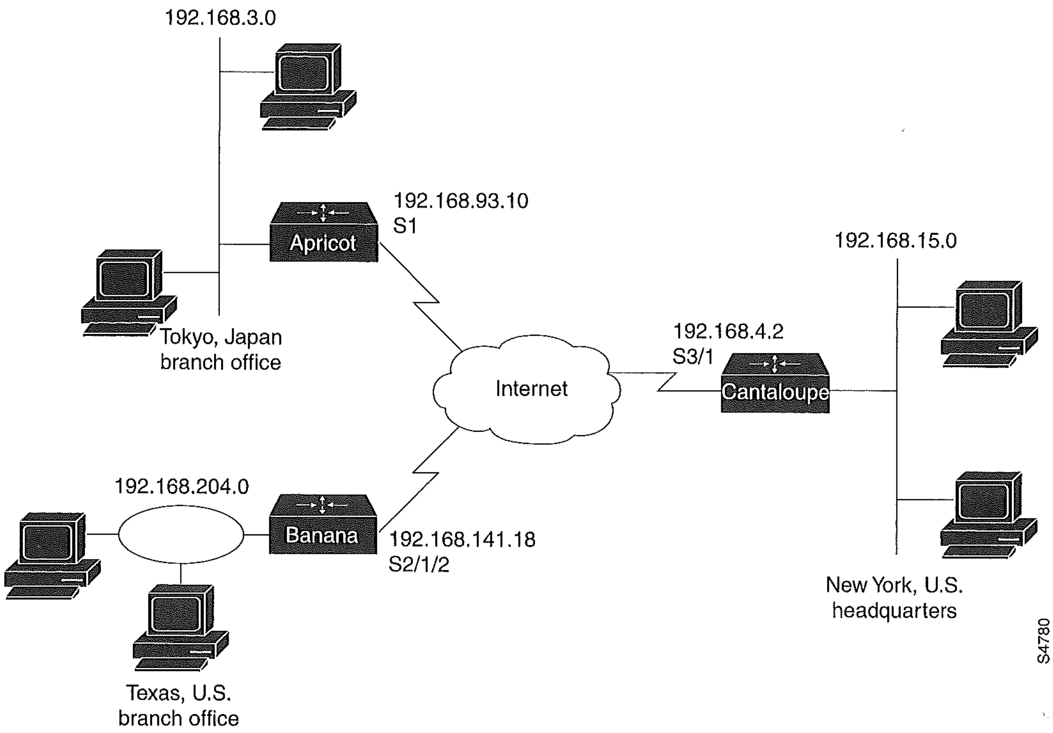
```
Banana(config)# access-list 110 permit tcp host 192.168.15.6 192.168.3.0 0.0.0.15
Banana(config)# crypto map Rsrch 10
Banana(config-crypto-map)# set peer Apricot
Banana(config-crypto-map)# set algorithm des cfb-8
Banana(config-crypto-map)# set algorithm des cfb-64
Banana(config-crypto-map)# match address 110
Banana(config-crypto-map)# exit
Banana(config)# interface s3/0/2
Banana(config-if)# crypto map Rsrch
Banana(config-if)# exit
Banana(config)#
```

Because Banana sets two DES algorithms for crypto map Rsrch, Banana could use either algorithm with traffic on the S3/0/2 interface. However, because Apricot only sets one DES algorithm (CFB-8 DES) for the crypto map Research, that is the only DES algorithm that will be used for all encrypted traffic between Apricot and Banana.

Example 2

In the second example, employees at two branch offices and at headquarters must communicate sensitive information. A mix of TCP and UDP traffic is transmitted by IP packets. Figure 32 shows the network topology used in this example.

Figure 32 Example 2 Network Topology



Apricot is a Cisco 2500 series router and connects to the Internet through interface S1. Both Banana and Cantaloupe are Cisco 7500 series routers with ESA cards. Banana connects to the Internet using the ESA-governed VIP2 interface S2/1/2. Cantaloupe is already using every VIP2 interface (governed by the ESA card) to connect to several offsite financial services, so it must connect to the Internet using a serial interface (S3/1) in slot 3. (Cantaloupe's interface S3/1 is governed by the Cisco IOS crypto engine.)

Apricot will be using one interface to communicate with both Banana and Cantaloupe. Because only one crypto map can be applied to this interface, Apricot creates a crypto map that has two distinct definition sets by using two different *seq-num* values with the same *map-name*. By using *seq-num* values of 10 and 20, Apricot creates a single crypto map set named "TXandNY" that contains a subset of definitions for encrypted sessions with Banana, and a second distinct subset for definitions for encrypted sessions with Cantaloupe.

Banana and Cantaloupe each also use a single interface to communicate with the other two routers, and therefore will use the same strategy as Apricot does for creating crypto map sets.

In this example, we assume that Apricot has generated DSS keys with the *key-name* "Apricot.TokyoBranch," Banana has generated DSS keys with the *key-name* "BananaESA.TXbranch," and Cantaloupe has generated DSS keys with the *key-name* "CantaloupeIOS.NY." We also assume that each router has exchanged DSS public keys with the other two routers, and that each router has enabled each DES algorithm that is specified in the crypto maps.

Apricot

```
Apricot(config)# access-list 105 permit tcp 192.168.3.0 0.0.0.15 192.168.204.0 0.0.0.255
Apricot(config)# access-list 105 permit udp 192.168.3.0 0.0.0.15 192.168.204.0 0.0.0.255
Apricot(config)# access-list 106 permit tcp 192.168.3.0 0.0.0.15 192.168.15.0 0.0.0.255
Apricot(config)# access-list 106 permit udp 192.168.3.0 0.0.0.15 192.168.15.0 0.0.0.255
Apricot(config)# crypto map TXandNY 10
Apricot(config-crypto-map)# set peer BananaESA.TXbranch
Apricot(config-crypto-map)# set algorithm 40-bit-des cfb-8
Apricot(config-crypto-map)# match address 105
Apricot(config-crypto-map)# exit
Apricot(config)# crypto map TXandNY 20
Apricot(config-crypto-map)# set peer CantaloupeIOS.NY
Apricot(config-crypto-map)# set algorithm 40-bit-des cfb-64
Apricot(config-crypto-map)# match address 106
Apricot(config-crypto-map)# exit
Apricot(config)# interface s1
Apricot(config-if)# crypto map TXandNY
Apricot(config-if)# exit
Apricot(config)#
```

Banana

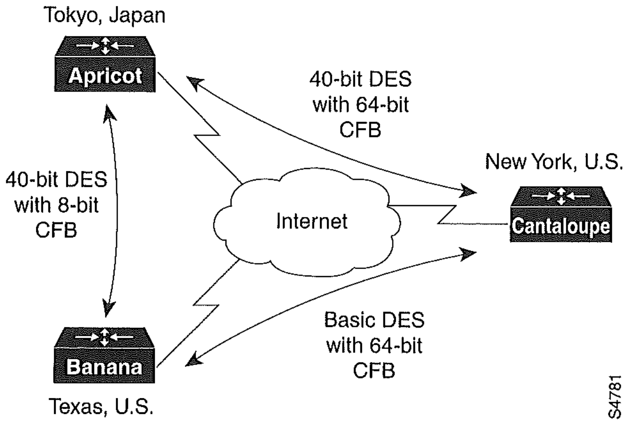
```
Banana(config)# access-list 110 permit tcp 192.168.204.0 0.0.0.255 192.168.3.0 0.0.0.15
Banana(config)# access-list 110 permit udp 192.168.204.0 0.0.0.255 192.168.3.0 0.0.0.15
Banana(config)# access-list 120 permit tcp 192.168.204.0 0.0.0.255 192.168.15.0 0.0.0.255
Banana(config)# access-list 120 permit udp 192.168.204.0 0.0.0.255 192.168.15.0 0.0.0.255
Banana(config)# crypto map USA 10
Banana(config-crypto-map)# set peer Apricot.TokyoBranch
Banana(config-crypto-map)# set algorithm 40-bit-des cfb-8
Banana(config-crypto-map)# match address 110
Banana(config-crypto-map)# exit
Banana(config)# crypto map USA 20
Banana(config-crypto-map)# set peer CantaloupeIOS.NY
Banana(config-crypto-map)# set algorithm des cfb-64
Banana(config-crypto-map)# match address 120
Banana(config-crypto-map)# exit
Banana(config)# interface s2/1/2
Banana(config-if)# crypto map USA
Banana(config-if)# exit
Banana(config)#
```

Cantaloupe

```
Cantaloupe(config)# access-list 101 permit tcp 192.168.15.0 0.0.0.255 192.168.3.0 0.0.0.15
Cantaloupe(config)# access-list 101 permit udp 192.168.15.0 0.0.0.255 192.168.3.0 0.0.0.15
Cantaloupe(config)# access-list 102 permit tcp 192.168.15.0 0.0.0.255 192.168.204.0 0.0.0.255
Cantaloupe(config)# access-list 102 permit udp 192.168.15.0 0.0.0.255 192.168.204.0 0.0.0.255
Cantaloupe(config)# crypto map satellites 10
Cantaloupe(config-crypto-map)# set peer Apricot.TokyoBranch
Cantaloupe(config-crypto-map)# set algorithm 40-bit-des cfb-64
Cantaloupe(config-crypto-map)# match address 101
Cantaloupe(config-crypto-map)# exit
Cantaloupe(config)# crypto map satellites 20
Cantaloupe(config-crypto-map)# set peer BananaESA.TXbranch
Cantaloupe(config-crypto-map)# set algorithm des cfb-64
Cantaloupe(config-crypto-map)# match address 102
Cantaloupe(config-crypto-map)# exit
Cantaloupe(config)# interface s3/1
Cantaloupe(config-if)# crypto map satellites
Cantaloupe(config-if)# exit
Cantaloupe(config)#
```

The previous configurations will result in DES encryption algorithms being applied to encrypted IP traffic as shown in Figure 33.

Figure 33 Example 2 DES Encryption Algorithms


Tokyo, Japan
Apricot
40-bit DES with 64-bit CFB
New York, U.S.
40-bit DES with 8-bit CFB
Internet
Cantaloupe
Basic DES with 64-bit CFB
Banana
Texas, U.S.
S4781

Changing Encryption Access List Limits Example

In this example, there are 50 remote sites connecting to a single server. The connections between the server and each site need to be encrypted. The server is located behind the local router named Apricot. Each of the remote sites connects through its own router.

Because of the large number of destination addresses that must be paired with the same source address in the local encryption access list, the default limits are changed.

```
Apricot(config)# crypto cisco connections 60
%Please reboot for the new connection size to take effect

Apricot(config)# crypto cisco entities 5
%Please reboot for the new table size to take effect
```

Even though there is only one server, and only 50 remote sites, this example defines 5 sources and 60 destinations. This allows room for future growth of the encryption access list. If another source or destination is added later, the limits will not have to be increased and the router rebooted again, which is a disruptive process.

Configuring Encryption with GRE Tunnels Examples

There are two example configurations for encryption with GRE tunnels:

- Example of Encrypting Only GRE Tunnel Traffic
- Example of Encrypting Both GRE Tunnel Traffic and Other Non-GRE Traffic

Example of Encrypting Only GRE Tunnel Traffic

This configuration causes all traffic through the GRE tunnel to be encrypted. No other traffic at the interface will be encrypted. The GRE tunnel is from router Apricot to router Banana. (Only partial configuration files are shown for each router.)

Apricot

```
crypto map BananaMap 10
 set algorithm 40-bit-des
 set peer Banana
 match address 101
!
interface Tunnel0
 no ip address
 ipx network 923FA800
 tunnel source 10.1.1.2
 tunnel destination 10.1.1.1
 crypto map BananaMap
!
interface Serial0
 ip address 10.1.1.2 255.255.255.0
 crypto map BananaMap
!
access-list 101 permit gre host 10.1.1.2 host 10.1.1.1
```

Banana

```
crypto map ApricotMap 10
 set algorithm 40-bit-des
 set peer Apricot
 match address 102
!
interface Tunnel0
 no ip address
 ipx network 923FA800
 tunnel source 10.1.1.1
 tunnel destination 10.1.1.2
 crypto map ApricotMap
!
interface Serial0
 ip address 10.1.1.1 255.255.255.0
 clockrate 2000000
 no cdp enable
 crypto map ApricotMap
!
access-list 102 permit gre host 10.1.1.1 host 10.1.1.2
```

Example of Encrypting Both GRE Tunnel Traffic and Other Non-GRE Traffic

This configuration encrypts all GRE tunnel traffic, and it also encrypts TCP traffic between two hosts with the IP addresses 172.16.25.3 and 192.168.3.5. The GRE tunnel is from router Apricot to router Banana. (Only partial configuration files are shown for each router.)

Apricot

```
crypto map BananaMapTunnel 10
 set algorithm 40-bit-des
 set peer Banana
 match address 101
!
crypto map BananaMapSerial 10
 set algorithm 40-bit-des
 set peer Banana
 match address 101
crypto map BananaMapSerial 20
 set algorithm 40-bit-des
 set peer Banana
 match address 110
!
interface Tunnel0
 no ip address
 ipx network 923FA800
 tunnel source 10.1.1.2
 tunnel destination 10.1.1.1
 crypto map BananaMapTunnel
!
interface Serial0
 ip address 10.1.1.2 255.255.255.0
 crypto map BananaMapSerial
!
access-list 101 permit gre host 10.1.1.2 host 10.1.1.1
access-list 110 permit tcp host 172.16.25.3 host 192.168.3.5
```

Banana

```
crypto map ApricotMapTunnel 10
 set algorithm 40-bit-des
 set peer Apricot
 match address 102
!
crypto map ApricotMapSerial 10
 set algorithm 40-bit-des
 set peer Apricot
 match address 102
crypto map ApricotMapSerial 20
 set algorithm 40-bit-des
 set peer Apricot
 match address 112
!
interface Tunnel0
 no ip address
 ipx network 923FA800
 tunnel source 10.1.1.1
 tunnel destination 10.1.1.2
 crypto map ApricotMapTunnel
!
interface Serial0
 ip address 10.1.1.1 255.255.255.0
 clockrate 2000000
 no cdp enable
 crypto map ApricotMapSerial
!
access-list 102 permit gre host 10.1.1.1 host 10.1.1.2
access-list 112 permit tcp host 192.168.3.5 host 172.16.25.3
```

ESA-Specific Encryption Configuration Task Examples

This section includes examples of the following:

- Examples of Resetting an ESA
- Example of Enabling an ESA (Cisco 7200 Series Routers Only)
- Examples of Selecting a Different Crypto Engine (Cisco 7200 Series Routers Only)

Examples of Resetting an ESA

The following example resets an ESA on a Cisco 7500 series router. The ESA is in a VIP2 that is in slot 4 of the router chassis.

```
Banana(config)# crypto card clear-latch 4
% Enter the crypto card password.
Password: <passwd>
Banana(config)#
```

The following example resets an ESA without DSS keys, for a Cisco 7200 series router. The ESA is in the router chassis slot 2.

```
Apricot(config)# crypto card clear-latch 2
% Enter the crypto card password.
Password: <passwd>
ESA in slot 2 not enabled.
[OK]
Apricot(config)#
```

The following example resets an ESA with DSS keys, for a Cisco 7200 series router; the ESA was previously in use on the same router, but was removed and reinstalled. No changes to the encryption configuration are desired by the administrator. The ESA is in the router chassis slot 2.

```
Apricot(config)# crypto card clear-latch 2
% Enter the crypto card password.
Password: <passwd>
Keys were found for this ESA- enable ESA now? [yes/no]: yes
...switching to HW crypto engine
[OK]
Apricot(config)#
```

The following example resets an ESA with DSS keys, for a Cisco 7200 series router; the ESA was previously used in a different router, and requires new DSS keys to be generated and exchanged before the ESA can become operational. The ESA is in the router chassis slot 2.

```
Apricot(config)# crypto card clear-latch 2
% Enter the crypto card password.
Password: <passwd>
Keys were found for this ESA- enable ESA now? [yes/no]: no
ESA in slot 2 not enabled.
[OK]
Apricot(config)#
```

Example of Enabling an ESA (Cisco 7200 Series Routers Only)

The following example enables an ESA in the router chassis slot 2:

```
Apricot(config)# crypto card enable 2
...switching to HW crypto engine
Apricot(config)#
```

Examples of Selecting a Different Crypto Engine (Cisco 7200 Series Routers Only)

Select a different crypto engine only if the new engine is fully configured for encryption.

The following example switches from the Cisco IOS crypto engine to the ESA crypto engine. The ESA crypto engine is in the router chassis slot 4.

```
Apricot(config)# crypto card enable 4
...switching to HW crypto engine
Apricot(config)#
```

The following example switches from the ESA crypto engine to the Cisco IOS crypto engine. The ESA crypto engine is in the router chassis slot 4.

```
Apricot(config)# crypto card shutdown 4
...switching to SW crypto engine
Apricot(config)#
```

Deleting DSS Keys Example

This section includes an example for a Cisco 7500 series router and an example for a Cisco 7200 series router with an installed ESA.

Example for a Cisco 7500 Series Router

The following example deletes all the DSS keys on a Cisco 7500 series router. The RSP is in chassis slot 3 and a VIP2 is in chassis slot 4. Deleting all the DSS keys turns off encryption completely for the router. The Cisco IOS crypto engine keys are deleted first, then the VIP2 crypto engine keys.

```
Apricot(config)# crypto key zeroize dss 3
Warning! Zeroize will remove your DSS signature keys.
Do you want to continue? [yes/no]: y
Keys to be removed are named Apricot.IOS.
Do you really want to remove these keys? [yes/no]: y
[OK]
Apricot(config)# crypto key zeroize dss 4
Warning! Zeroize will remove your DSS signature keys.
Do you want to continue? [yes/no]: y
Keys to be removed are named Apricot.VIP.
Do you really want to remove these keys? [yes/no]: y
[OK]
Apricot(config)#
```

Example for a Cisco 7200 Series Router

The following example deletes DSS keys only for an ESA, in chassis slot 2 of a Cisco 7200 series router. The Cisco IOS crypto engine DSS keys are not deleted in this example.

1 View existing DSS keys:

```
Apricot# show crypto key mypubkey dss
crypto key pubkey-chain dss Apricot.IOS 01709642
BDD99A6E EEE53D30 BC0BFAE6 948C40FB 713510CB 32104137 91B06C8D C2D5B422
D9C154CA 00CDE99B 425DB9FD FE3162F1 1E5866AF CF66DD33 677259FF E5C24812
quit
crypto key pubkey-chain dss Apricot.ESA 01234567
866AFCF6 E99B425D FDFE3162 BC0BFAE6 13791B06 713510CB 4CA00CDE 0BC0BFAE
3791B06C 154C0CDE F11E5866 AE6948C4 DD336772 3F66DF33 355459FF 2350912D
quit

Apricot#
```

This output shows that DSS keys exist for both the Cisco IOS crypto engine and for the ESA crypto engine.

2 Determine the Active Crypto Engine:

```
Apricot# show crypto engine configuration
engine name:        Apricot.IOS
engine type:        software
serial number:      01709642
platform:           rsp crypto engine

Encryption Process Info:
input queue top:    44
input queue bot:    44
input queue count:  0

Apricot#
```

The output shows that the Cisco IOS crypto engine is the active engine.

3 Because we want to delete DSS keys for the ESA crypto engine, change to the ESA crypto engine:

```
Apricot# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Apricot(config)# crypto card enable 2
...switching to HW crypto engine
Apricot(config)#
```

4 Verify that the ESA crypto engine is the active engine:

```
Apricot(config)# exit
Apricot# show crypto engine configuration
engine name:        Apricot.ESA
engine type:        hardware
serial number:      01234567
platform:           esa crypto engine

Encryption Process Info:
input queue top:    0
input queue bot:    0
input queue count:  0

Apricot#
```

The output shows that the ESA crypto engine is now the active engine.

5. Delete the ESA DSS keys:

```
Apricot# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Apricot(config)# crypto key zeroize dss 2
Warning! Zeroize will remove your DSS signature keys.
Do you want to continue? [yes/no]: y
Keys to be removed are named Apricot.ESA.
Do you really want to remove these keys? [yes/no]: y
[OK]

Apricot(config)#
```

6. View existing DSS keys:

```
Apricot(config)# exit
Apricot# show crypto key mypubkey dss
crypto key pubkey-chain dss Apricot.IOS 01709642
BDD99A6E EEE53D30 BC0BFAE6 948C40FB 713510CB 32104137 91B06C8D C2D5B422
D9C154CA 00CDE99B 425DB9FD FE3162F1 1E5866AF CF66DD33 677259FF E5C24812
quit

Apricot#
```

The output shows that the ESA crypto engine keys have been deleted.

7. Determine the active crypto engine:

```
Apricot# show crypto engine configuration
engine name:        Apricot.IOS
engine type:        software
serial number:      01709642
platform:           rsp crypto engine

Encryption Process Info:
input queue top:     0
input queue bot:     0
input queue count:   0

Apricot#
```

The output shows that the system has defaulted back to the Cisco IOS crypto engine as the active engine.

Testing the Encryption Connection Example

The following example sets up and verifies a test encryption session.

Assume the same network topology and configuration as in the previous example and shown in Figure 32.

In this example, router Apricot sets up a test encryption session with router Banana and then views the connection status to verify a successful encrypted session connection.

Step 1 Router Apricot sets up a test encryption connection with router Banana.

```
Apricot# test crypto initiate-session 192.168.3.12 192.168.204.110
BananaESA.TXbranch 10
Sending CIM to: 192.168.204.110 from: 192.168.3.12.
Connection id: -1
```

Notice the Connection id value is –1. A negative value indicates that the connection is being set up.

Step 2 Router Apricot issues the **show crypto cisco connections** command.

```
Apricot# show crypto cisco connections
Pending Connection Table
PE              UPE              Timestamp              Conn_id
192.168.3.10    192.168.204.100 Mar 01 1993 00:01:09   -1

Connection Table
PE              UPE              Conn_id New_id  Alg     Time
192.168.3.10    192.168.204.100 -1       1       0       Not Set
                flags:PEND_CONN
```

Look in the Pending Connection Table for an entry with a Conn_id value equal to the previously shown Connection id value—in this case, look for an entry with a Conn_id value of –1. If this is the first time an encrypted connection has been attempted, there will only be one entry (as shown).

Note the PE and UPE addresses for this entry.

Step 3 Now, look in the Connection Table for an entry with the same PE and UPE addresses. In this case, there is only one entry in both tables, so finding the right Connection Table entry is easy.

Step 4 At the Connection Table entry, note the Conn_id and New_id values. In this case, Conn_id equals –1 (as in the Pending Connection Table), and New_id equals 1. The New_id value of 1 will be assigned to the test connection when setup is complete. (Positive numbers are assigned to established, active connections.)

Step 5 Apricot waits a few seconds for the test connection to establish and then reissues the **show crypto cisco connections** command.

```
Apricot# show crypto cisco connections
Connection Table
PE              UPE              Conn_id New_id  Alg     Time
192.168.3.10    192.168.204.100 1        0       0       Mar 01 1993 00:02:00
                flags:TIME_KEYS
```

Again, look for the Connection Table entry with the same PE and UPE addresses as shown before. In this entry, notice that the Conn_id value has changed to 1. This indicates that our test connection has been successfully established, because the Conn_id value has changed to match the New_id value of Step 4. Also, New_id has been reset to 0 at this point, indicating that there are no new connections currently being set up.

In the command output of Step 5, there is no longer a Pending Connection Table being displayed, which indicates that there are currently no pending connections. This is also a good clue that the test connection was successfully established.

The **show crypto cisco connections** command is explained in greater detail in the chapter "Cisco Encryption Technology Commands" in the *Cisco IOS Security Command Reference*. There you can find a description of how connection IDs are assigned during and following connection setup.

Configuring IPSec Network Security

This chapter describes how to configure IPSec, which is a framework of open standards developed by the Internet Engineering Task Force (IETF). IPSec provides security for transmission of sensitive information over unprotected networks such as the Internet. IPSec acts at the network layer, protecting and authenticating IP packets between participating IPSec devices ("peers"), such as Cisco routers.

IPSec provides the following network security services. These services are optional. In general, local security policy will dictate the use of one or more of these services:

- Data Confidentiality—The IPSec sender can encrypt packets before transmitting them across a network.
- Data Integrity—The IPSec receiver can authenticate packets sent by the IPSec sender to ensure that the data has not been altered during transmission.
- Data Origin Authentication—The IPSec receiver can authenticate the source of the IPSec packets sent. This service is dependent upon the data integrity service.
- Anti-Replay—The IPSec receiver can detect and reject replayed packets.

Note The term *data authentication* is generally used to mean data integrity and data origin authentication. Within this chapter it also includes anti-replay services, unless otherwise specified.

With IPSec, data can be transmitted across a public network without fear of observation, modification, or spoofing. This enables applications such as Virtual Private Networks (VPNs), including intranets, extranets, and remote user access.

For a complete description of the IPSec Network Security commands used in this chapter, refer to the "IPSec Network Security Commands" chapter in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

In This Chapter

This chapter has the following sections:

- IPSec Overview
- IPSec Configuration Task List
- IPSec Configuration Example

IPSec Overview

IPSec provides network data encryption at the IP packet level, offering a robust security solution that is standards-based. IPSec provides data authentication and anti-replay services in addition to data confidentiality services.

This section has the following sections:

- Supported Standards
- List of Terms
- Supported Hardware, Switching Paths, and Encapsulation
- Restrictions
- Overview of How IPSec Works
- Nesting of IPSec Traffic to Multiple Peers
- Prerequisites

Supported Standards

Cisco implements the following standards with this feature:

- **IPSec**—IP Security Protocol. IPSec is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer; it uses IKE to handle negotiation of protocols and algorithms based on local policy, and to generate the encryption and authentication keys to be used by IPSec. IPSec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

  Note The term IPSec is sometimes used to describe the entire protocol of IPSec data services and IKE security protocols and is also sometimes used to describe only the data services.

  IPSec is documented in a series of Internet Drafts, all available at http://www.ietf.org/html.charters/ipsec-charter.html. The overall IPSec implementation is per the latest version of the "Security Architecture for the Internet Protocol" Internet Draft (RFC2401). Cisco IOS IPSec implements RFC 2402 (IP Authentication Header) though RFC 2410 (The NULL Encryption Algorithm and Its Use With IPSec).

- **Internet Key Exchange (IKE)**—A hybrid protocol which implements Oakley and SKEME key exchanges inside the ISAKMP framework. While IKE can be used with other protocols, its initial implementation is with the IPSec protocol. IKE provides authentication of the IPSec peers, negotiates IPSec security associations, and establishes IPSec keys.

  For more information on IKE, see the "Configuring Internet Key Exchange Security Protocol" chapter.

The component technologies implemented for IPSec include:

- **DES**—The Data Encryption Standard (DES) is used to encrypt packet data. Cisco IOS implements the mandatory 56-bit DES-CBC with Explicit IV. Cipher Block Chaining (CBC) requires an initialization vector (IV) to start encryption. The IV is explicitly given in the IPSec packet. For backwards compatibility, Cisco IOS IPSec also implements the RFC 1829 version of ESP DES-CBC.

Cisco IOS also implements Triple DES (168-bit) encryption, depending on the software versions available for a specific platform. Triple DES (3DES) is a strong form of encryption that allows sensitive information to be transmitted over untrusted networks. It enables customers to utilize network layer encryption.

Caution Cisco IOS images with strong encryption (including, but not limited to, 56-bit data encryption feature sets) are subject to United States government export controls, and have a limited distribution. Images to be installed outside the United States require an export license. Customer orders might be denied or subject to delay due to United States government regulations. Contact your sales representative or distributor for more information, or send e-mail to export@cisco.com.

- **MD5 (HMAC variant)**—MD5 (Message Digest 5) is a hash algorithm. HMAC is a keyed hash variant used to authenticate data.
- **SHA (HMAC variant)**—SHA (Secure Hash Algorithm) is a hash algorithm. HMAC is a keyed hash variant used to authenticate data.

IPSec as implemented in Cisco IOS software supports the following additional standards:

- **AH**—Authentication Header. A security protocol which provides data authentication and optional anti-replay services. AH is embedded in the data to be protected (a full IP datagram).
- **ESP**—Encapsulating Security Payload. A security protocol which provides data privacy services and optional data authentication, and anti-replay services. ESP encapsulates the data to be protected.

List of Terms

anti-replay—A security service where the receiver can reject old or duplicate packets in order to protect itself against replay attacks. IPSec provides this optional service by use of a sequence number combined with the use of data authentication. Cisco IOS IPSec provides this service whenever it provides the data authentication service, except in the following cases:

The service is not available for manually established security associations (that is, security associations established by configuration and not by IKE).

data authentication—Includes two concepts:

- Data integrity (verify that data has not been altered).
- Data origin authentication (verify that the data was actually sent by the claimed sender).

Data authentication can refer either to integrity alone or to both of these concepts (although data origin authentication is dependent upon data integrity).

data confidentiality—A security service where the protected data cannot be observed.

data flow—A grouping of traffic, identified by a combination of source address/mask, destination address/mask, IP next protocol field, and source and destination ports, where the protocol and port fields can have the values of **any**. In effect, all traffic matching a specific combination of these values is logically grouped together into a data flow. A data flow can represent a single TCP connection between two hosts, or it can represent all of the traffic between two subnets. IPSec protection is applied to data flows.

peer—In the context of this chapter, a peer refers to a router or other device that participates in IPSec.

perfect forward secrecy (PFS)—A cryptographic characteristic associated with a derived shared secret value. With PFS, if one key is compromised, previous and subsequent keys are not compromised, because subsequent keys are not derived from previous keys.

security association—An IPSec security association (SA) is a description of how two or more entities will use security services in the context of a particular security protocol (AH or ESP) to communicate securely on behalf of a particular data flow. It includes such things as the transform and the shared secret keys to be used for protecting the traffic.

The IPSec security association is established either by IKE or by manual user configuration. Security associations are unidirectional and are unique per security protocol. So when security associations are established for IPSec, the security associations (for each protocol) for both directions are established at the same time.

When using IKE to establish the security associations for the data flow, the security associations are established when needed and expire after a period of time (or volume of traffic). If the security associations are manually established, they are established as soon as the necessary configuration is completed and do not expire.

security parameter index (SPI)—This is a number which, together with a destination IP address and security protocol, uniquely identifies a particular security association. When using IKE to establish the security associations, the SPI for each security association is a pseudo-randomly derived number. Without IKE, the SPI is manually specified for each security association.

transform—The list of operations done on a dataflow to provide data authentication, data confidentiality, and data compression. For example, one transform is the ESP protocol with the HMAC-MD5 authentication algorithm; another transform is the AH protocol with the 56-bit DES encryption algorithm and the ESP protocol with the HMAC-SHA authentication algorithm.

tunnel—In the context of this chapter, a secure communication path between two peers, such as two routers. It does not refer to using IPSec in tunnel mode.

Supported Hardware, Switching Paths, and Encapsulation

IPSec has certain restrictions for hardware, switching paths, and encapsulation methods as follows:

Supported Hardware

For 7100, 7200, and 7500 hardware platforms, IPSec support requires the following adaptors or modules:

- Integrated Services Adapter (ISA) for the Cisco 7200 or 7500 series.
- Integrated Services Modules (ISM) for the Cisco 7100 series.

Supported Switching Paths

IPSec works with process switching, fast switching, and Cisco Express Forwarding (CEF). IPSec does not work with optimum or flow switching.

Supported Encapsulation

IPSec works with the following serial encapsulations: High-Level Data-Links Control (HDLC), Point-to-Point Protocol (PPP), and Frame Relay.

IPSec also works with the GRE and IPinIP Layer 3, L2F, L2TP, DLSw+, and SRB tunneling protocols; however, multipoint tunnels are not supported. Other Layer 3 tunneling protocols may not be supported for use with IPSec.

Because the IPSec Working Group has not yet addressed the issue of group key distribution, IPSec currently cannot be used to protect group traffic (such as broadcast or multicast traffic).

Restrictions

At this time, IPSec can be applied to unicast IP datagrams only. Because the IPSec Working Group has not yet addressed the issue of group key distribution, IPSec does not currently work with multicasts or broadcast IP datagrams.

If you use Network Address Translation (NAT), you should configure static NAT translations so that IPSec will work properly. In general, NAT translation should occur before the router performs IPSec encapsulation; in other words, IPSec should be working with global addresses.

Overview of How IPSec Works

In simple terms, IPSec provides secure *tunnels* between two peers, such as two routers. You define which packets are considered sensitive and should be sent through these secure tunnels, and you define the parameters which should be used to protect these sensitive packets, by specifying characteristics of these tunnels. Then, when the IPSec peer sees such a sensitive packet, it sets up the appropriate secure tunnel and sends the packet through the tunnel to the remote peer.

Note The use of the term *tunnel* in this chapter does not refer to using IPSec in tunnel mode.

More accurately, these *tunnels* are sets of security associations that are established between two IPSec peers. The security associations define which protocols and algorithms should be applied to sensitive packets, and also specify the keying material to be used by the two peers. Security associations are unidirectional and are established per security protocol (AH or ESP).

With IPSec you define what traffic should be protected between two IPSec peers by configuring access lists and applying these access lists to interfaces by way of crypto map sets. Therefore, traffic may be selected based on source and destination address, and optionally Layer 4 protocol, and port. (The access lists used for IPSec are used only to determine which traffic should be protected by IPSec, not which traffic should be blocked or permitted through the interface. Separate access lists define blocking and permitting at the interface.)

A crypto map set can contain multiple entries, each with a different access list. The crypto map entries are searched in order—the router attempts to match the packet to the access list specified in that entry.

When a packet matches a **permit** entry in a particular access list, and the corresponding crypto map entry is tagged as **cisco**, and connections are established if necessary. If the crypto map entry is tagged as **ipsec-isakmp**, IPSec is triggered. If no security association exists that IPSec can use to protect this traffic to the peer, IPSec uses IKE to negotiate with the remote peer to set up the necessary IPSec security associations on behalf of the data flow. The negotiation uses information specified in the crypto map entry as well as the data flow information from the specific access list entry. (The behavior is different for dynamic crypto map entries. Refer to the "Creating Dynamic Crypto Maps" section.)

If the crypto map entry is tagged as **ipsec-manual**, IPSec is triggered. If no security association exists that IPSec can use to protect this traffic to the peer, the traffic is dropped. In this case, the security associations are installed via the configuration, without the intervention of IKE. If the security associations did not exist, IPSec did not have all of the necessary pieces configured.

Once established, the set of security associations (outbound, to the peer) is then applied to the triggering packet as well as to subsequent applicable packets as those packets exit the router. "Applicable" packets are packets that match the same access list criteria that the original packet matched. For example, all applicable packets could be encrypted before being forwarded to the remote peer. The corresponding inbound security associations are used when processing the incoming traffic from that peer.

If IKE is used to establish the security associations, the security associations will have lifetimes so that they will periodically expire and require renegotiation. (This provides an additional level of security.)

Multiple IPSec tunnels can exist between two peers to secure different data streams, with each tunnel using a separate set of security associations. For example, some data streams might be just authenticated while other data streams must both be encrypted and authenticated.

Access lists associated with IPSec crypto map entries also represent which traffic the router requires to be protected by IPSec. Inbound traffic is processed against the crypto map entries—if an unprotected packet matches a **permit** entry in a particular access list associated with an IPSec crypto map entry, that packet is dropped because it was not sent as an IPSec-protected packet.

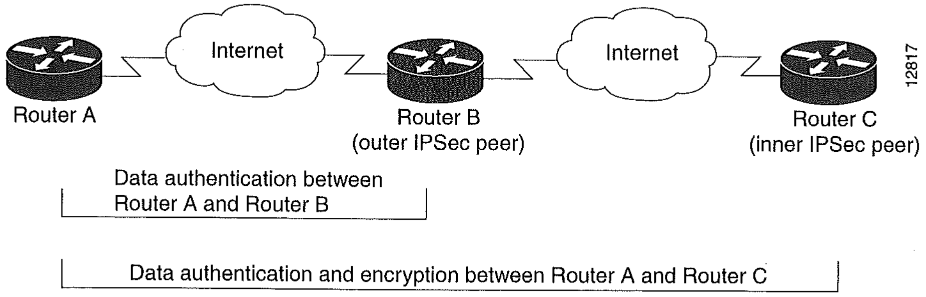
Crypto map entries also include transform sets. A transform set is an acceptable combination of security protocols, algorithms and other settings to apply to IPSec protected traffic. During the IPSec security association negotiation, the peers agree to use a particular transform set when protecting a particular data flow.

Nesting of IPSec Traffic to Multiple Peers

You can nest IPSec traffic to a series of IPSec peers. For example, in order for traffic to traverse multiple firewalls (and these firewalls have a policy of not letting through traffic that they themselves have not authenticated), the router needs to establish IPSec tunnels with each firewall in turn. The "nearest" firewall becomes the "outermost" IPSec peer.

In the example shown in Figure 34, Router A encapsulates the traffic destined for Router C in IPSec (Router C is the IPSec peer). However, before Router A can send this traffic, it must first reencapsulate this traffic in IPSec in order to send it to Router B (Router B is the "outermost" IPSec peer).

Figure 34 Nesting Example of IPSec Peers



It is possible for the traffic between the "outer" peers to have one kind of protection (such as data authentication) and for traffic between the "inner" peers to have different protection (such as both data authentication and encryption).

Prerequisites

You need to configure IKE as described in the "Configuring Internet Key Exchange Security Protocol" chapter.

Even if you decide to not use IKE, you still need to disable it as described in the "Configuring Internet Key Exchange Security Protocol" chapter.

IPSec Configuration Task List

After you have completed IKE configuration, configure IPSec. To configure IPSec, perform the tasks in the following sections at each participating IPSec peer.

- Ensuring Access Lists Are Compatible with IPSec
- Setting Global Lifetimes for IPSec Security Associations
- Creating Crypto Access Lists
- Defining Transform Sets
- Creating Crypto Map Entries
- Applying Crypto Map Sets to Interfaces
- Monitoring and Maintaining IPSec

For IPSec configuration examples, refer to the "IPSec Configuration Example" section at the end of this chapter.

Ensuring Access Lists Are Compatible with IPSec

IKE uses UDP port 500. The IPSec ESP and AH protocols use protocol numbers 50 and 51. Ensure that your access lists are configured so that protocol 50, 51, and UDP port 500 traffic is not blocked at interfaces used by IPSec. In some cases you might need to add a statement to your access lists to explicitly permit this traffic.

Setting Global Lifetimes for IPSec Security Associations

You can change the global lifetime values which are used when negotiating new IPSec security associations. (These global lifetime values can be overridden for a particular crypto map entry).

These lifetimes only apply to security associations established via IKE. Manually established security associations do not expire.

There are two lifetimes: a "timed" lifetime and a "traffic-volume" lifetime. A security association expires after the first of these lifetimes is reached. The default lifetimes are 3600 seconds (one hour) and 4,608,000 kilobytes (10 megabytes per second for one hour).

If you change a global lifetime, the new lifetime value will not be applied to currently existing security associations, but will be used in the negotiation of subsequently established security associations. If you wish to use the new values immediately, you can clear all or part of the security association database. Refer to the **clear crypto sa** command for more details.

IPSec security associations use one or more shared secret keys. These keys and their security associations time out together.

To change a global lifetime for IPSec security associations, use one or more of the following commands in global configuration mode:

Command	Purpose
crypto ipsec security-association lifetime seconds *seconds*	Changes the global "timed" lifetime for IPSec SAs. This command causes the security association to time out after the specified number of seconds have passed.
crypto ipsec security-association lifetime kilobytes *kilobytes*	Changes the global "traffic-volume" lifetime for IPSec SAs. This command causes the security association to time out after the specified amount of traffic (in kilobytes) have passed through the IPSec "tunnel" using the security association.
clear crypto sa or **clear crypto sa peer** {*ip-address* | *peer-name*} or **clear crypto sa map** *map-name* or **clear crypto sa entry** *destination-address protocol spi*	(Optional) Clears existing security associations. This causes any existing security associations to expire immediately; future security associations will use the new lifetimes. Otherwise, any existing security associations will expire according to the previously configured lifetimes. **Note** Using the **clear crypto sa** command without parameters will clear out the full SA database, which will clear out active security sessions. You may also specify the **peer**, **map**, or **entry** keywords to clear out only a subset of the SA database. For more information, see the **clear crypto sa** command.

How These Lifetimes Work

Assuming that the particular crypto map entry does not have lifetime values configured, when the router requests new security associations it will specify its global lifetime values in the request to the peer; it will use this value as the lifetime of the new security associations. When the router receives a negotiation request from the peer, it will use the smaller of either the lifetime value proposed by the peer or the locally configured lifetime value as the lifetime of the new security associations.

The security association (and corresponding keys) will expire according to whichever comes sooner, either after the number of seconds has passed (specified by the **seconds** keyword) or after the amount of traffic in kilobytes is passed (specified by the **kilobytes** keyword). Security associations that are established manually (via a crypto map entry marked as **ipsec-manual**) have an infinite lifetime.

A new security association is negotiated *before* the lifetime threshold of the existing security association is reached, to ensure that a new security association is ready for use when the old one expires. The new security association is negotiated either 30 seconds before the **seconds** lifetime expires or when the volume of traffic through the tunnel reaches 256 kilobytes less than the **kilobytes** lifetime (whichever comes first).

If no traffic has passed through the tunnel during the entire life of the security association, a new security association is not negotiated when the lifetime expires. Instead, a new security association will be negotiated only when IPSec sees another packet that should be protected.

Creating Crypto Access Lists

Crypto access lists are used to define which IP traffic will be protected by crypto and which traffic will not be protected by crypto. (These access lists are *not* the same as regular access lists, which determine what traffic to forward or block at an interface.) For example, access lists can be created to protect all IP traffic between Subnet A and Subnet Y or Telnet traffic between Host A and Host B.

The access lists themselves are not specific to IPSec. It is the crypto map entry referencing the specific access list that defines whether IPSec processing is applied to the traffic matching a **permit** in the access list.

Crypto access lists associated with IPSec crypto map entries have four primary functions:

- Select outbound traffic to be protected by IPSec (permit = protect).
- Indicate the data flow to be protected by the new security associations (specified by a single **permit** entry) when initiating negotiations for IPSec security associations.
- Process inbound traffic in order to filter out and discard traffic that should have been protected by IPSec.
- Determine whether or not to accept requests for IPSec security associations on behalf of the requested data flows when processing IKE negotiation from the IPSec peer. (Negotiation is only done for **ipsec-isakmp** crypto map entries.) In order to be accepted, if the peer initiates the IPSec negotiation, it must specify a data flow that is "permitted" by a crypto access list associated with an **ipsec-isakmp** crypto map entry.

If you want certain traffic to receive one combination of IPSec protection (for example, authentication only) and other traffic to receive a different combination of IPSec protection (for example, both authentication and encryption), you need to create two different crypto access lists to define the two different types of traffic. These different access lists are then used in different crypto map entries which specify different IPSec policies.

Later, you will associate the crypto access lists to particular interfaces when you configure and apply crypto map sets to the interfaces (following instructions in the sections "Creating Crypto Map Entries" and "Applying Crypto Map Sets to Interfaces").

To create crypto access lists, use the following command in global configuration mode:

Command	Purpose
access-list *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**log**] or **ip access-list extended** *name* Follow with **permit** and **deny** statements as appropriate.	Specifies conditions to determine which IP packets will be protected.[1] (Enable or disable crypto for traffic that matches these conditions.) Cisco recommends that you configure "mirror image" crypto access lists for use by IPSec and that you avoid using the **any** keyword, as described in the sections "Defining Mirror Image Crypto Access Lists at Each IPSec Peer" and "Using the any Keyword in Crypto Access Lists" (following). Also see the "Crypto Access List Tips" section.

[1] You specify conditions using an IP access list designated by either a number or a name. The **access-list** command designates a numbered extended access list; the **ip access-list extended** command designates a named access list.

This section has the following sections:

- Crypto Access List Tips
- Defining Mirror Image Crypto Access Lists at Each IPSec Peer
- Using the any Keyword in Crypto Access Lists

Crypto Access List Tips

Using the **permit** keyword causes all IP traffic that matches the specified conditions to be protected by crypto, using the policy described by the corresponding crypto map entry. Using the **deny** keyword prevents traffic from being protected by crypto in the context of that particular crypto map entry. (In other words, it does not allow the policy as specified in this crypto map entry to be applied to this traffic.) If this traffic is denied in all of the crypto map entries for that interface, then the traffic is not protected by crypto.

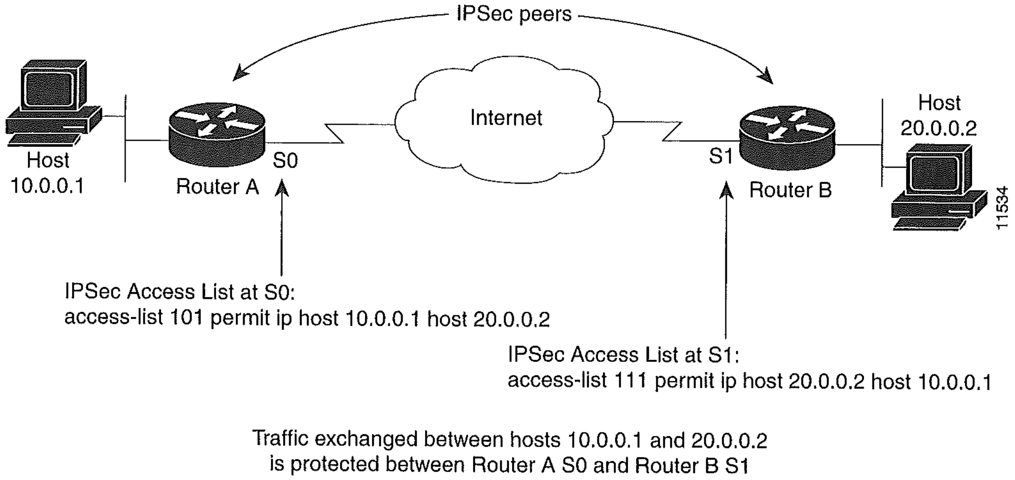
The crypto access list you define will be applied to an interface after you define the corresponding crypto map entry and apply the crypto map set to the interface. Different access lists must be used in different entries of the same crypto map set. (These two tasks are described in following sections.) However, both inbound and outbound traffic will be evaluated against the same "outbound" IPSec access list. Therefore, the access list's criteria is applied in the forward direction to traffic exiting your router, and the reverse direction to traffic entering your router. In Figure 35, IPSec protection is applied to traffic between Host 10.0.0.1 and Host 20.0.0.2 as the data exits Router A's S0 interface enroute to Host 20.0.0.2. For traffic from Host 10.0.0.1 to Host 20.0.0.2, the access list entry on Router A is evaluated as follows:

```
source = host 10.0.0.1
dest = host 20.0.0.2
```

For traffic from Host 20.0.0.2 to Host 10.0.0.1, that same access list entry on Router A is evaluated as follows:

```
source = host 20.0.0.2
dest = host 10.0.0.1
```

Figure 35 How Crypto Access Lists Are Applied for Processing IPSec


IPSec peers
Internet
Host
10.0.0.1
Router A
S0
S1
Router B
Host
20.0.0.2
11534
IPSec Access List at S0:
access-list 101 permit ip host 10.0.0.1 host 20.0.0.2
IPSec Access List at S1:
access-list 111 permit ip host 20.0.0.2 host 10.0.0.1
Traffic exchanged between hosts 10.0.0.1 and 20.0.0.2
is protected between Router A S0 and Router B S1

If you configure multiple statements for a given crypto access list which is used for IPSec, in general the first **permit** statement that is matched will be the statement used to determine the scope of the IPSec security association. That is, the IPSec security association will be set up to protect traffic that meets the criteria of the matched statement only. Later, if traffic matches a different **permit** statement of the crypto access list, a new, separate IPSec security association will be negotiated to protect traffic matching the newly matched access list statement.

Note Access lists for crypto map entries tagged as **ipsec-manual** are restricted to a single **permit** entry and subsequent entries are ignored. In other words, the security associations established by that particular crypto map entry are only for a single data flow. To be able to support multiple manually established security associations for different kinds of traffic, define multiple crypto access lists, and then apply each one to a separate **ipsec-manual** crypto map entry. Each access list should include one **permit** statement defining what traffic to protect.

Any unprotected inbound traffic that matches a **permit** entry in the crypto access list for a crypto map entry flagged as IPSec will be dropped, because this traffic was expected to be protected by IPSec.

Note If you view your router's access lists by using a command such as **show ip access-lists**, *all* extended IP access lists will be shown in the command output. This includes extended IP access lists that are used for traffic filtering purposes as well as those that are used for crypto. The **show** command output does not differentiate between the different uses of the extended access lists.

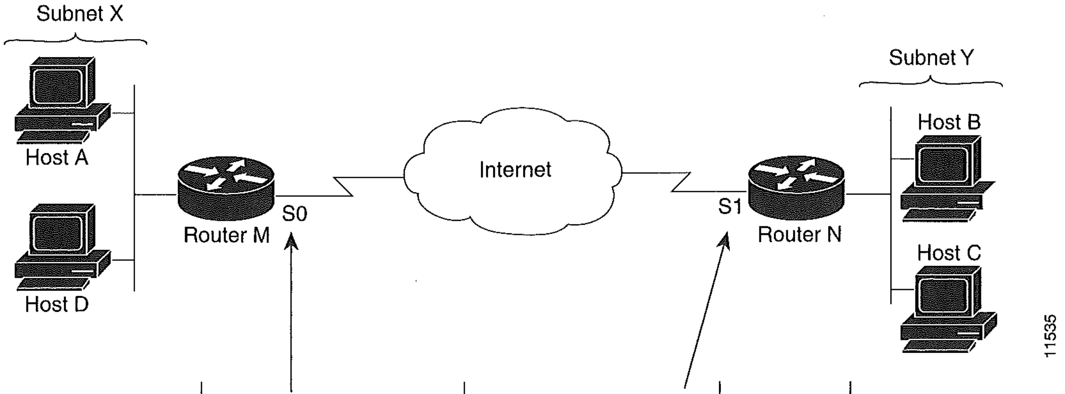
See the *Cisco IOS Security Command Reference* for complete details about the extended IP access list commands used to create IPSec access lists.

Defining Mirror Image Crypto Access Lists at Each IPSec Peer

Cisco recommends that for every crypto access list specified for a static crypto map entry that you define at the local peer, you define a "mirror image" crypto access list at the remote peer. This ensures that traffic that has IPSec protection applied locally can be processed correctly at the remote peer. (The crypto map entries themselves must also support common transforms and must refer to the other system as a peer.)

Figure 36 shows some sample scenarios when you have mirror image access lists and when you do not have mirror image access lists.

Figure 36 Mirror Image vs. Non-Mirror Image Crypto Access Lists (for IPSec)



		IPSec Access List at S0:	IPSec Access List at S1:	1st Packet	Result
Mirror image access lists at Router M S0 and Router N S1	Case 1	permits Host A → Host B	permits Host B → Host A	A → B or B → A	SAs established for traffic A↔ B (good)
	Case 2	permits Subnet X → Subnet Y	permits Subnet Y → Subnet X	A → B or B → A or A → C or C → D etc.	SAs established for traffic X↔Y (good)
	Case 3	permits Host A → Host B	permits Subnet Y → Subnet X	A → B	SAs established for traffic A↔B (good)
	Case 4			B → A	SAs cannot be established and packets from Host B to Host A are dropped (bad)

As Figure 36 indicates, IPSec Security Associations (SAs) can be established as expected whenever the two peers' crypto access lists are mirror images of each other. However, an IPSec SA can be established only some of the time when the access lists are not mirror images of each other. This can happen in the case where an entry in one peer's access list is a subset of an entry in the other peer's access list, such as shown in Cases 3 and 4 of Figure 36. IPSec SA establishment is critical to IPSec—without SAs, IPSec does not work, causing any packets matching the crypto access list criteria to be silently dropped instead of being forwarded with IPSec security.

In Figure 36, an SA cannot be established in Case 4. This is because SAs are always requested according to the crypto access lists at the initiating packet's end. In Case 4, Router B requests that all traffic between Subnet X and Subnet Y be protected, but this is a superset of the specific flows permitted by the crypto access list at Router A so the request is therefore not permitted. Case 3 works because Router A's request is a subset of the specific flows permitted by the crypto access list at Router B.

Because of the complexities introduced when crypto access lists are not configured as mirror images at peer IPSec devices, Cisco strongly encourages you to use mirror image crypto access lists.

Using the **any** Keyword in Crypto Access Lists

When you create crypto access lists, using the **any** keyword could cause problems. Cisco discourages the use of the **any** keyword to specify source or destination addresses.

The **any** keyword in a **permit** statement is discouraged when you have multicast traffic flowing through the IPSec interface; the **any** keyword can cause multicast traffic to fail.

The **permit any any** statement is strongly discouraged, as this will cause all outbound traffic to be protected (and all protected traffic sent to the peer specified in the corresponding crypto map entry) and will require protection for all inbound traffic. Then, all inbound packets that lack IPSec protection will be silently dropped, including packets for routing protocols, NTP, echo, echo response, and so on.

You need to be sure you define which packets to protect. If you *must* use the **any** keyword in a **permit** statement, you must preface that statement with a series of **deny** statements to filter out any traffic (that would otherwise fall within that **permit** statement) that you do not want to be protected.

Defining Transform Sets

A transform set represents a certain combination of security protocols and algorithms. During the IPSec security association negotiation, the peers agree to use a particular transform set for protecting a particular data flow.

You can specify multiple transform sets, and then specify one or more of these transform sets in a crypto map entry. The transform set defined in the crypto map entry will be used in the IPSec security association negotiation to protect the data flows specified by that crypto map entry's access list.

During IPSec security association negotiations with IKE, the peers search for a transform set that is the same at both peers. When such a transform set is found, it is selected and will be applied to the protected traffic as part of both peers' IPSec security associations.

With manually established security associations, there is no negotiation with the peer, so both sides must specify the same transform set.

If you change a transform set definition, the change is only applied to crypto map entries that reference the transform set. The change will not be applied to existing security associations, but will be used in subsequent negotiations to establish new security associations. If you want the new settings to take effect sooner, you can clear all or part of the security association database by using the **clear crypto sa** command.

To define a transform set, use the following commands starting in global configuration mode:

Step	Command	Purpose
1	**crypto ipsec transform-set** *transform-set-name transform1* [*transform2* [*transform3*]]	Defines a transform set. There are complex rules defining which entries you can use for the transform arguments. These rules are explained in the command description for the **crypto ipsec transform-set** command, and Table 25 provides a list of allowed transform combinations. This command puts you into the crypto transform configuration mode.
2	**mode** [**tunnel** | **transport**]	(Optional) Changes the mode associated with the transform set. The mode setting is only applicable to traffic whose source and destination addresses are the IPSec peer addresses; it is ignored for all other traffic. (All other traffic is in tunnel mode only.)
3	**exit**	Exits the crypto transform configuration mode.
4	**clear crypto sa** or **clear crypto sa peer** {*ip-address* | *peer-name*} or **clear crypto sa map** *map-name* or **clear crypto sa entry** *destination-address protocol spi*	Clears existing IPSec security associations so that any changes to a transform set will take effect on subsequently established security associations. (Manually established SAs are reestablished immediately.) Using the **clear crypto sa** command without parameters will clear out the full SA database, which will clear out active security sessions. You may also specify the **peer**, **map**, or **entry** keywords to clear out only a subset of the SA database. For more information, see the **clear crypto sa** command.

Table 25 shows allowed transform combinations.

Table 25 Allowed Transform Combinations

Transform Type	Transform	Description
AH Transform *(Pick up to one.)*		
	ah-md5-hmac	AH with the MD5 (HMAC variant) authentication algorithm
	ah-sha-hmac	AH with the SHA (HMAC variant) authentication algorithm
	ah-sha-hmac	AH with the SHA (HMAC variant) authentication algorithm
ESP Encryption Transform *(Pick up to one.)*		
	esp-des	ESP with the 56-bit DES encryption algorithm
	esp-3des	ESP with the 168-bit DES encryption algorithm (3DES or Triple DES)
	esp-null	Null encryption algorithm

Table 25 Allowed Transform Combinations (continued)

Transform Type	Transform	Description
ESP Authentication Transform *(Pick up to one.)*		
	esp-md5-hmac	ESP with the MD5 (HMAC variant) authentication algorithm
	esp-sha-hmac	ESP with the SHA (HMAC variant) authentication algorithm
IP Compression Transform *(Pick up to one.)*		
	comp-lzs	IP compression with the LZS algorithm.

Creating Crypto Map Entries

To create crypto map entries, follow the guidelines and tasks described in these sections:

- About Crypto Maps
- Load Sharing
- How Many Crypto Maps Should You Create?
- Creating Crypto Map Entries to Establish Manual Security Associations
- Creating Crypto Map Entries that Use IKE to Establish Security Associations
- Creating Dynamic Crypto Maps

About Crypto Maps

Crypto map entries created for IPSec pull together the various parts used to set up IPSec security associations, including:

- Which traffic should be protected by IPSec (per a crypto access list)
- The granularity of the flow to be protected by a set of security associations
- Where IPSec-protected traffic should be sent (who the remote IPSec peer is)
- The local address to be used for the IPSec traffic (See the "Applying Crypto Map Sets to Interfaces" section for more details.)
- What IPSec security should be applied to this traffic (selecting from a list of one or more transform sets)
- Whether security associations are manually established or are established via IKE
- Other parameters that might be necessary to define an IPSec security association

Crypto map entries with the same crypto map name (but different map sequence numbers) are grouped into a crypto map set. Later, you will apply these crypto map sets to interfaces; then, all IP traffic passing through the interface is evaluated against the applied crypto map set. If a crypto map entry sees outbound IP traffic that should be protected and the crypto map specifies the use of IKE, a security association is negotiated with the remote peer according to the parameters included in the crypto map entry; otherwise, if the crypto map entry specifies the use of manual security associations, a security association should have already been established via configuration. (If a dynamic crypto map entry sees outbound traffic that should be protected and no security association exists, the packet is dropped.)

The policy described in the crypto map entries is used during the negotiation of security associations. If the local router initiates the negotiation, it will use the policy specified in the static crypto map entries to create the offer to be sent to the specified IPSec peer. If the IPSec peer initiates the negotiation, the local router will check the policy from the static crypto map entries, as well as any referenced dynamic crypto map entries to decide whether to accept or reject the peer's request (offer).

For IPSec to succeed between two IPSec peers, both peers' crypto map entries must contain compatible configuration statements.

When two peers try to establish a security association, they must each have at least one crypto map entry that is compatible with one of the other peer's crypto map entries. For two crypto map entries to be compatible, they must at least meet the following criteria:

- The crypto map entries must contain compatible crypto access lists (for example, mirror image access lists). In the case where the responding peer is using dynamic crypto maps, the entries in the local crypto access list must be "permitted" by the peer's crypto access list.
- The crypto map entries must each identify the other peer (unless the responding peer is using dynamic crypto maps).
- The crypto map entries must have at least one transform set in common.

Load Sharing

You can define multiple remote peers using crypto maps to allow for load sharing. If one peer fails, there will still be a protected path. The peer that packets are actually sent to is determined by the last peer that the router heard from (received either traffic or a negotiation request from) for a given data flow. If the attempt fails with the first peer, IKE tries the next peer on the crypto map list.

If you are not sure how to configure each crypto map parameter to guarantee compatibility with other peers, you might consider configuring dynamic crypto maps as described in the "Creating Dynamic Crypto Maps" section. Dynamic crypto maps are useful when the establishment of the IPSec tunnels is initiated by the remote peer (such as in the case of an IPSec router fronting a server). They are not useful if the establishment of the IPSec tunnels is locally initiated, because the dynamic crypto maps are policy templates, not complete statements of policy. (Although the access lists in any referenced dynamic crypto map entry are used for crypto packet filtering.)

How Many Crypto Maps Should You Create?

You can apply only one crypto map set to a single interface. The crypto map set can include a combination of IPSec/IKE and IPSec/manual entries. Multiple interfaces can share the same crypto map set if you want to apply the same policy to multiple interfaces.

If you create more than one crypto map entry for a given interface, use the *seq-num* of each map entry to rank the map entries: the lower the *seq-num*, the higher the priority. At the interface that has the crypto map set, traffic is evaluated against higher priority map entries first.

You must create multiple crypto map entries for a given interface if any of the following conditions exist:

- If different data flows are to be handled by separate IPSec peers.
- If you want to apply different IPSec security to different types of traffic (to the same or separate IPSec peers); for example, if you want traffic between one set of subnets to be authenticated, and traffic between another set of subnets to be both authenticated and encrypted. In this case the different types of traffic should have been defined in two separate access lists, and you must create a separate crypto map entry for each crypto access list.
- If you are not using IKE to establish a particular set of security associations, and want to specify multiple access list entries, you must create separate access lists (one per **permit** entry) and specify a separate crypto map entry for each access list.

Creating Crypto Map Entries to Establish Manual Security Associations

The use of manual security associations is a result of a prior arrangement between the users of the local router and the IPSec peer. The two parties may wish to begin with manual security associations, and then move to using security associations established via IKE, or the remote party's system may not support IKE. If IKE is not used for establishing the security associations, there is no negotiation of security associations, so the configuration information in both systems must be the same in order for traffic to be processed successfully by IPSec.

The local router can simultaneously support manual and IKE-established security associations, even within a single crypto map set. There is very little reason to disable IKE on the local router (unless the router only supports manual security associations, which is unlikely).

To create crypto map entries to establish manual security associations (SAs) (that is, when IKE is not used to establish the SAs), use the following commands starting in global configuration mode:

Step	Command	Purpose
1	**crypto map** *map-name seq-num* **ipsec-manual**	Specifies the crypto map entry to create (or modify). This command puts you into the crypto map configuration mode.
2	**match address** *access-list-id*	Names an IPSec access list. This access list determines which traffic should be protected by IPSec and which traffic should not be protected by IPSec security in the context of this crypto map entry. (The access list can specify only one **permit** entry when IKE is not used.)
3	**set peer** {*hostname* | *ip-address*}	Specifies the remote IPSec peer. This is the peer to which IPSec protected traffic should be forwarded. (Only one peer can be specified when IKE is not used.)
4	**set transform-set** *transform-set-name*	Specifies which transform set should be used. This must be the same transform set that is specified in the remote peer's corresponding crypto map entry. (Only one transform set can be specified when IKE is not used.)
5	**set session-key inbound ah** *spi hex-key-string* and **set session-key outbound ah** *spi hex-key-string*	If the specified transform set includes the AH protocol, sets the AH Security Parameter Indexes (SPIs) and keys to apply to inbound and outbound protected traffic. (This manually specifies the AH security association to be used with protected traffic.)

Step	Command	Purpose
6	**set session-key inbound esp** *spi* **cipher** *hex-key-string* [**authenticator** *hex-key-string*] and **set session-key outbound esp** *spi* **cipher** *hex-key-string* [**authenticator** *hex-key-string*]	If the specified transform set includes the ESP protocol, sets the ESP Security Parameter Indexes (SPIs) and keys to apply to inbound and outbound protected traffic. If the transform set includes an ESP cipher algorithm, specify the cipher keys. If the transform set includes an ESP authenticator algorithm, specify the authenticator keys. (This manually specifies the ESP security association to be used with protected traffic.)
7	**exit**	Exits crypto-map configuration mode and return to global configuration mode.

Repeat these steps to create additional crypto map entries as required.

Creating Crypto Map Entries that Use IKE to Establish Security Associations

When IKE is used to establish security associations, the IPSec peers can negotiate the settings they will use for the new security associations. This means that you can specify lists (such as lists of acceptable transforms) within the crypto map entry.

To create crypto map entries that will use IKE to establish the security associations, use the following commands starting in global configuration mode:

Step	Command	Purpose
1	**crypto map** *map-name seq-num* **ipsec-isakmp**	Names the crypto map entry to create (or modify). This command puts you into the crypto map configuration mode.
2	**match address** *access-list-id*	Names an extended access list. This access list determines which traffic should be protected by IPSec and which traffic should not be protected by IPSec security in the context of this crypto map entry.
3	**set peer** {*hostname* | *ip-address*}	Specifies a remote IPSec peer. This is the peer to which IPSec protected traffic can be forwarded. Repeat for multiple remote peers.
4	**set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*]	Specifies which transform sets are allowed for this crypto map entry. List multiple transform sets in order of priority (highest priority first).
5	**set security-association lifetime seconds** *seconds* and/or **set security-association lifetime kilobytes** *kilobytes*	(Optional) If you want the security associations for this crypto map entry to be negotiated using different IPSec security association lifetimes than the global lifetimes, specify a security association lifetime for the crypto map entry.
6	**set security-association level per-host**	(Optional) Specifies that separate security associations should be established for each source/destination host pair. Without this command, a single IPSec "tunnel" could carry traffic for multiple source hosts and multiple destination hosts. With this command, when the router requests new security associations it will establish one set for traffic between Host A and Host B, and a separate set for traffic between Host A and Host C. Use this command with care, as multiple streams between given subnets can rapidly consume resources.

Step	Command	Purpose
7	`set pfs [group1	group2]`	(Optional) Specifies that IPSec should ask for perfect forward secrecy when requesting new security associations for this crypto map entry, or should demand PFS in requests received from the IPSec peer.
8	`exit`	Exits crypto-map configuration mode and return to global configuration mode.

Repeat these steps to create additional crypto map entries as required.

Creating Dynamic Crypto Maps

Dynamic crypto maps (this requires IKE) can ease IPSec configuration and are recommended for use with networks where the peers are not always predetermined. An example of this is mobile users, who obtain dynamically-assigned IP addresses. First, the mobile clients need to authenticate themselves to the local router's IKE by something other than an IP address, such as a fully qualified domain name. Once authenticated, the security association request can be processed against a dynamic crypto map which is set up to accept requests (matching the specified local policy) from previously unknown peers.

To configure dynamic crypto maps, follow the instructions in these sections:

- Understanding Dynamic Crypto Maps
- Creating a Dynamic Crypto Map Set
- Adding the Dynamic Crypto Map Set into a Regular (Static) Crypto Map Set

Understanding Dynamic Crypto Maps

Dynamic crypto maps are only available for use by IKE.

A dynamic crypto map entry is essentially a crypto map entry without all the parameters configured. It acts as a policy template where the missing parameters are later dynamically configured (as the result of an IPSec negotiation) to match a remote peer's requirements. This allows remote peers to exchange IPSec traffic with the router even if the router does not have a crypto map entry specifically configured to meet all of the remote peer's requirements.

Dynamic crypto maps are not used by the router to initiate new IPSec security associations with remote peers. Dynamic crypto maps are used when a remote peer tries to initiate an IPSec security association with the router. Dynamic crypto maps are also used in evaluating traffic.

A dynamic crypto map set is included by reference as part of a crypto map set. Any crypto map entries that reference dynamic crypto map sets should be the lowest priority crypto map entries in the crypto map set (that is, have the highest sequence numbers) so that the other crypto map entries are evaluated first; that way, the dynamic crypto map set is examined only when the other (static) map entries are not successfully matched.

If the router accepts the peer's request, at the point that it installs the new IPSec security associations it also installs a temporary crypto map entry. This entry is filled in with the results of the negotiation. At this point, the router performs normal processing, using this temporary crypto map entry as a normal entry, even requesting new security associations if the current ones are expiring (based upon the policy specified in the temporary crypto map entry). Once the flow expires (that is, all of the corresponding security associations expire), the temporary crypto map entry is then removed.

For both static and dynamic crypto maps, if unprotected inbound traffic matches a **permit** statement in an access list, and the corresponding crypto map entry is tagged as "IPSec," then the traffic is dropped because it is not IPSec-protected. (This is because the security policy as specified by the crypto map entry states that this traffic must be IPSec-protected.)

For static crypto map entries, if outbound traffic matches a **permit** statement in an access list and the corresponding SA is not yet established, the router will initiate new SAs with the remote peer. In the case of dynamic crypto map entries, if no SA existed, the traffic would simply be dropped (because dynamic crypto maps are not used for initiating new SAs).

Note Use care when using the **any** keyword in **permit** entries in dynamic crypto maps. If it is possible for the traffic covered by such a **permit** entry to include multicast or broadcast traffic, the access list should include **deny** entries for the appropriate address range. Access lists should also include **deny** entries for network and subnet broadcast traffic, and for any other traffic that should not be IPSec protected.

Creating a Dynamic Crypto Map Set

Dynamic crypto map entries, like regular static crypto map entries, are grouped into sets. A set is a group of dynamic crypto map entries all with the same *dynamic-map-name* but each with a different *dynamic-seq-num*.

To create a dynamic crypto map entry, use the following commands starting in global configuration mode:

Step	Command	Purpose
1	**crypto dynamic-map** *dynamic-map-name dynamic-seq-num*	Creates a dynamic crypto map entry.
2	**set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*]	Specifies which transform sets are allowed for the crypto map entry. List multiple transform sets in order of priority (highest priority first). This is the only configuration statement required in dynamic crypto map entries.
3	**match address** *access-list-id*	(Optional) Accesses list number or name of an extended access list. This access list determines which traffic should be protected by IPSec and which traffic should not be protected by IPSec security in the context of this crypto map entry. **Note** Although access-lists are optional for dynamic crypto maps, they are highly recommended If this is configured, the data flow identity proposed by the IPSec peer must fall within a **permit** statement for this crypto access list. If this is not configured, the router will accept any data flow identity proposed by the IPSec peer. However, if this is configured but the specified access list does not exist or is empty, the router will drop all packets. This is similar to static crypto maps because they also require that an access list be specified. Care must be taken if the **any** keyword is used in the access list, because the access list is used for packet filtering as well as for negotiation.
4	**set peer** {*hostname* | *ip-address*}	(Optional) Specifies a remote IPSec peer. Repeat for multiple remote peers. This is rarely configured in dynamic crypto map entries. Dynamic crypto map entries are often used for unknown remote peers.

Step	Command	Purpose
5	**set security-association lifetime seconds** *seconds* and/or **set security-association lifetime kilobytes** *kilobytes*	(Optional) If you want the security associations for this crypto map to be negotiated using shorter IPSec security association lifetimes than the globally specified lifetimes, specify a key lifetime for the crypto map entry.
6	**set pfs** [**group1** | **group2**]	(Optional) Specifies that IPSec should ask for perfect forward secrecy when requesting new security associations for this crypto map entry or should demand perfect forward secrecy in requests received from the IPSec peer.
7	**exit**	Exits crypto-map configuration mode and return to global configuration mode.

Dynamic crypto map entries specify crypto access lists that limit traffic for which IPSec security associations can be established. A dynamic crypto map entry that does not specify an access list will be ignored during traffic filtering. A dynamic crypto map entry with an empty access list causes traffic to be dropped. If there is only one dynamic crypto map entry in the crypto map set, it must specify acceptable transform sets.

Adding the Dynamic Crypto Map Set into a Regular (Static) Crypto Map Set

You can add one or more dynamic crypto map sets into a crypto map set, via crypto map entries that reference the dynamic crypto map sets. You should set the crypto map entries referencing dynamic maps to be the lowest priority entries in a crypto map set (that is, have the highest sequence numbers).

To add a dynamic crypto map set into a crypto map set, use the following command in global configuration mode:

Command	Purpose
crypto map *map-name seq-num* **ipsec-isakmp dynamic** *dynamic-map-name*	Adds a dynamic crypto map set to a static crypto map set.

Applying Crypto Map Sets to Interfaces

You need to apply a crypto map set to each interface through which IPSec traffic will flow. Applying the crypto map set to an interface instructs the router to evaluate all the interface's traffic against the crypto map set and to use the specified policy during connection or security association negotiation on behalf of traffic to be protected by crypto.

To apply a crypto map set to an interface, use the following command in interface configuration mode:

Command	Purpose
crypto map *map-name*	Applies a crypto map set to an interface.

For redundancy, you could apply the same crypto map set to more than one interface. The default behavior is as follows:

- Each interface will have its own piece of the security association database.
- The IP address of the local interface will be used as the local address for IPSec traffic originating from or destined to that interface.

If you apply the same crypto map set to multiple interfaces for redundancy purposes, you need to specify an identifying interface. This has the following effects:

- The per-interface portion of the IPSec security association database will be established one time and shared for traffic through all the interfaces that share the same crypto map.
- The IP address of the identifying interface will be used as the local address for IPSec traffic originating from or destined to those interfaces sharing the same crypto map set.

One suggestion is to use a loopback interface as the identifying interface.

To specify redundant interfaces and name an identifying interface, use the following command in global configuration mode:

Command	Purpose
crypto map *map-name* **local-address** *interface-id*	Permits redundant interfaces to share the same crypto map, using the same local identity.

Monitoring and Maintaining IPSec

Certain configuration changes will only take effect when negotiating subsequent security associations. If you want the new settings to take immediate effect, you must clear the existing security associations so that they will be re-established with the changed configuration. For manually established security associations, you must clear and reinitialize the security associations or the changes will never take effect. If the router is actively processing IPSec traffic, it is desirable to clear only the portion of the security association database that would be affected by the configuration changes (that is, clear only the security associations established by a given crypto map set). Clearing the full security association database should be reserved for large-scale changes, or when the router is processing very little other IPSec traffic.

To clear (and reinitialize) IPSec security associations, use one of the following commands in global configuration mode:

Command	Purpose
clear crypto sa or **clear crypto sa peer** {*ip-address* | *peer-name*} or **clear crypto sa map** *map-name* or **clear crypto sa entry** *destination-address protocol spi*	Clears IPSec security associations. **Note** Using the **clear crypto sa** command without parameters will clear out the full SA database, which will clear out active security sessions. You may also specify the **peer**, **map**, or **entry** keywords to clear out only a subset of the SA database. For more information, see the **clear crypto sa** command.

To view information about your IPSec configuration, use one or more of the following commands in EXEC mode:

Command	Purpose
show crypto ipsec transform-set	Displays your transform set configuration.
show crypto map [**interface** *interface* | **tag** *map-name*]	Displays your crypto map configuration.
show crypto ipsec sa [**map** *map-name* | **address** | **identity**] [**detail**]	Displays information about IPSec security associations.
show crypto dynamic-map [**tag** *map-name*]	Displays information about dynamic crypto maps.
show crypto ipsec security-association lifetime	Displays global security association lifetime values.

IPSec Configuration Example

The following example shows a minimal IPSec configuration where the security associations will be established via IKE. For more information about IKE, see the "Configuring Internet Key Exchange Security Protocol" chapter.

An IPSec access list defines which traffic to protect:

```
access-list 101 permit ip 10.0.0.0 0.0.0.255 10.2.2.0 0.0.0.255
```

A transform set defines how the traffic will be protected. In this example, transform set "myset1" uses DES encryption and SHA for data packet authentication:

```
crypto ipsec transform-set myset1 esp-des esp-sha
```

Another transform set example is "myset2," which uses Triple DES encryptions and MD5 (HMAC variant) for data packet authentication:

```
crypto ipsec transform-set myset2 esp-3des esp-md5-hmac
```

A crypto map joins together the IPSec access list and transform set and specifies where the protected traffic is sent (the remote IPSec peer):

```
crypto map toRemoteSite 10 ipsec-isakmp
 match address 101
 set transform-set myset2
 set peer 10.2.2.5
```

The crypto map is applied to an interface:

```
interface Serial0
 ip address 10.0.0.2
 crypto map toRemoteSite
```

Note In this example, IKE must be enabled.

Configuring Certification Authority Interoperability

This chapter describes how to configure certification authority (CA) interoperability, which is provided in support of the IP Security (IPSec) protocol. CA interoperability permits Cisco IOS devices and CAs to communicate so that your Cisco IOS device can obtain and use digital certificates from the CA. Although IPSec can be implemented in your network without the use of a CA, using a CA provides manageability and scalability for IPSec.

For background and configuration information for IPSec, see the "Configuring IPSec Network Security" chapter.

For a complete description of the commands used in this chapter, refer to the "Certification Authority Interoperability Commands" chapter in the *Cisco IOS Security Command Reference*. To locate documentation for other commands that appear in this chapter, use the command reference master index or search online.

In This Chapter

This chapter includes the following sections:

- CA Interoperability Overview
- Overview of Certification Authorities
- CA Interoperability Configuration Task Lists
- What to Do Next
- CA Interoperability Configuration Examples

CA Interoperability Overview

Without CA interoperability, Cisco IOS devices could not use CAs when deploying IPSec. CAs provide a manageable, scalable solution for IPSec networks. For details, see the "Overview of Certification Authorities" section.

This section includes the following sections:

- Supported Standards
- Restrictions
- Prerequisites

Supported Standards

Cisco supports the following standards with this feature:

- **IPSec**—IP Security Protocol. IPSec is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer; it uses IKE to handle negotiation of protocols and algorithms based on local policy, and to generate the encryption and authentication keys to be used by IPSec. IPSec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

  For more information on IPSec, see the "Configuring IPSec Network Security" chapter.
- **Internet Key Exchange (IKE)**—A hybrid protocol which implements Oakley and Skeme key exchanges inside the ISAKMP framework. While IKE can be used with other protocols, its initial implementation is with the IPSec protocol. IKE provides authentication of the IPSec peers, negotiates IPSec keys, and negotiates IPSec security associations.

  For more information on IKE, see the "Configuring Internet Key Exchange Security Protocol" chapter.
- **Public-Key Cryptography Standard #7 (PKCS #7)**—A standard from RSA Data Security, Inc. used to encrypt and sign certificate enrollment messages.
- **Public-Key Cryptography Standard #10 (PKCS #10)**—A standard syntax from RSA Data Security, Inc. for certificate requests.
- **RSA Keys**—RSA is the public key cryptographic system developed by Ron Rivest, Adi Shamir, and Leonard Adleman. RSA keys come in pairs: one public key and one private key.
- **X.509v3 certificates**—Certificate support which allows the IPSec-protected network to scale by providing the equivalent of a digital ID card to each device. When two devices wish to communicate, they exchange digital certificates to prove their identity (thus removing the need to manually exchange public keys with each peer or to manually specify a shared key at each peer). These certificates are obtained from a certification authority (CA). X.509 is part of the X.500 standard by the ITU.

Restrictions

This feature should be configured only when you also configure both IPSec and IKE in your network.

Prerequisites

You need to have a certification authority (CA) available to your network before you configure this interoperability feature. The CA must support Cisco's PKI protocol, the certificate enrollment protocol (CEP).

Overview of Certification Authorities

This section provides background information about CAs, including the following:

- Purpose of CAs
- Implementing IPSec Without CAs
- Implementing IPSec with CAs

- How CA Certificates Are Used by IPSec Devices
- About Registration Authorities

Purpose of CAs

CAs are responsible for managing certificate requests and issuing certificates to participating IPSec network devices. These services provide centralized key management for the participating devices.

CAs simplify the administration of IPSec network devices. You can use a CA with a network containing multiple IPSec-compliant devices such as routers.

Digital signatures, enabled by public key cryptography, provide a means to digitally authenticate devices and individual users. In public key cryptography, such as the RSA encryption system, each user has a key-pair containing both a public and a private key. The keys act as complements, and anything encrypted with one of the keys can be decrypted with the other. In simple terms, a signature is formed when data is encrypted with a user's private key. The receiver verifies the signature by decrypting the message with the sender's public key. The fact that the message could be decrypted using the sender's public key indicates that the holder of the private key, the sender, must have created the message. This process relies on the receiver having a copy of the sender's public key and knowing with a high degree of certainty that it really does belong to the sender, and not to someone pretending to be the sender.

Digital certificates provide this link. A digital certificate contains information to identify a user or device, such as the name, serial number, company, department or IP address. It also contains a copy of the entity's public key. The certificate is itself signed by a certification authority (CA), a third party that is explicitly trusted by the receiver to validate identities and to create digital certificates.

In order to validate the CA's signature, the receiver must first know the CA's public key. Normally this is handled out-of-band or through an operation done at installation. For instance, most web browsers are configured with the public keys of several CAs by default. The Internet Key Exchange (IKE), a key component of IPSec, can use digital signatures to scalably authenticate peer devices before setting up security associations.

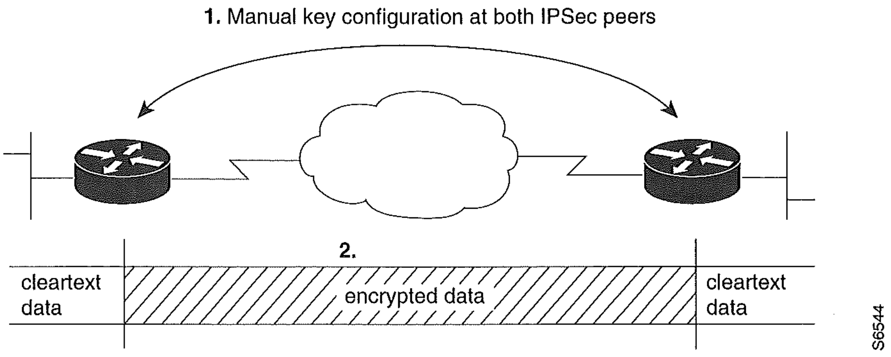
Without digital signatures, one must either manually exchange public keys or secrets between each pair of devices that use IPSec to protect communications between them. Without certificates, every new device added to the network requires a configuration change on every other device it securely communicates with. However, by using digital certificates, each device is enrolled with a certification authority. When two devices wish to communicate, they exchange certificates and digitally sign data to authenticate each other. When a new device is added to the network, one simply enrolls that device with a CA, and none of the other devices need modification. When the new device attempts an IPSec connection, certificates are automatically exchanged and the device can be authenticated.

Implementing IPSec Without CAs

Without a CA, if you want to enable IPSec services (such as encryption) between two Cisco routers, you must first ensure that each router has the other router's key (such as an RSA public key or a shared key). This requires that you manually perform one of the following:

- At each router, enter the other router's RSA public key
- At each router, specify a shared key to be used between the routers.

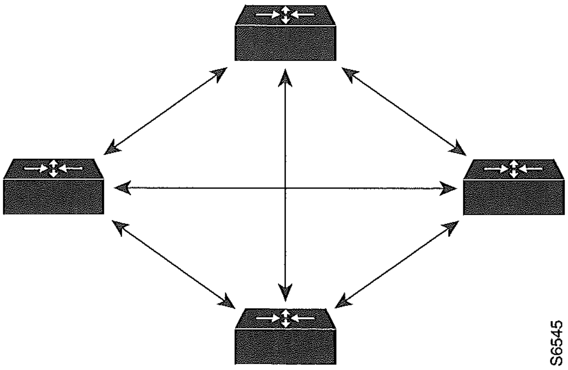
Figure 37 Without a CA: Key Configuration Between Two Routers



In Figure 37, each router uses the other router's key to authenticate the identity of the other router; this authentication always occurs whenever IPSec traffic is exchanged between the two routers.

If you have multiple Cisco routers in a mesh topology, and wish to exchange IPSec traffic passing between all of those routers, you must first configure shared keys or RSA public keys between all of those routers.

Figure 38 Without a CA: Six 2-Part Key Configurations Required for Four IPSec Routers



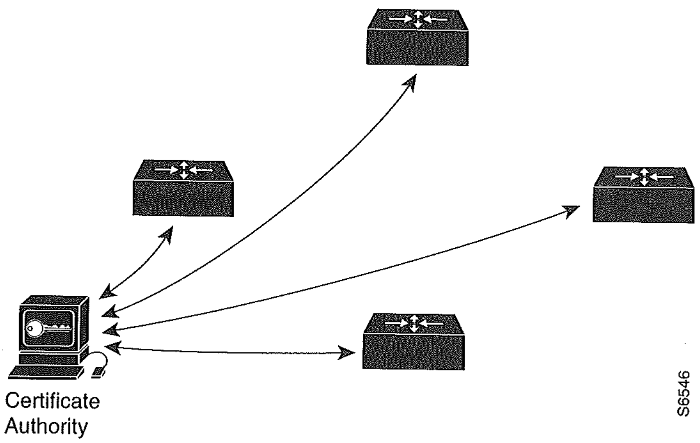
Every time a new router is added to the IPSec network, you must configure keys between the new router and each of the existing routers. (In Figure 38, four additional 2-part key configurations would be required to add a single encrypting router to the network.)

Consequently, the more devices there are that require IPSec services, the more involved the key administration becomes. Obviously, this approach does not scale well for larger, more complex encrypting networks.

Implementing IPSec with CAs

With a CA, you do not need to configure keys between all of the encrypting routers. Instead, you individually enroll each participating router with the CA, requesting a certificate for the router. When this has been accomplished, each participating router can dynamically authenticate all of the other participating routers. This is illustrated in Figure 39.

Figure 39 With a CA: Each Router Individually Makes Requests of the CA at Installation



To add a new IPSec router to the network, you only need to configure that new router to request a certificate from the CA, instead of making multiple key configurations with all the other existing IPSec routers.

How CA Certificates Are Used by IPSec Devices

When two IPSec routers want to exchange IPSec-protected traffic passing between them, they must first authenticate each other—otherwise, IPSec protection cannot occur. The authentication is done with IKE.

Without a CA, a router authenticates itself to the remote router using either RSA encrypted nonces or pre-shared keys. Both methods require that keys must have been previously configured between the two routers.

With a CA, a router authenticates itself to the remote router by sending a certificate to the remote router and performing some public key cryptography. Each router must send its own unique certificate that was issued and validated by the CA. This process works because each router's certificate encapsulates the router's public key, each certificate is authenticated by the CA, and all participating routers recognize the CA as an authenticating authority. This is called IKE with an RSA signature.

Your router can continue sending its own certificate for multiple IPSec sessions, and to multiple IPSec peers, until the certificate expires. When its certificate expires, the router administrator must obtain a new one from the CA.

CAs can also revoke certificates for devices that will no longer participate in IPSec. Revoked certificates are not recognized as valid by other IPSec devices. Revoked certificates are listed in a certificate revocation list (CRL), which each peer may check before accepting another peer's certificate.

About Registration Authorities

Some CAs have a Registration Authority (RA) as part of their implementation. An RA is essentially a server that acts as a proxy for the CA so that CA functions can continue when the CA is offline.

Some of the configuration tasks described in this document differ slightly depending on whether your CA supports an RA or not.

CA Interoperability Configuration Task Lists

To enable your Cisco device to interoperate with a CA, complete the tasks in the following sections. Some of the tasks are optional; the remaining are required.

- Managing NVRAM Memory Usage (Optional)
- Configuring the Router's Host Name and IP Domain Name (Required)
- Generating an RSA Key Pair (Required)
- Declaring a Certification Authority (Required)
- Authenticating the CA (Required)
- Requesting Your Own Certificate(s) (Required)
- Saving Your Configuration (Required)
- Monitoring and Maintaining Certification Authority Interoperability (Optional)

For CA interoperability configuration examples, refer to the "CA Interoperability Configuration Examples" section at the end of this chapter.

Managing NVRAM Memory Usage

Certificates and certificate revocation lists (CRLs) are used by your router when a CA is used. Normally certain certificates and all CRLs are stored locally in the router's NVRAM, and each certificate and CRL uses a moderate amount of memory.

- What certificates are normally stored at your router?
  - — Your router's certificate
  - — The CA's certificate
  - — Two Registration Authority (RA) certificates (only if the CA supports an RA)
- What CRLs are normally stored at your router?
  - — If your CA does not support an RA, only one CRL gets stored at your router.
  - — If your CA supports an RA, multiple CRLs can be stored at your router.

In some cases, storing these certificates and CRLs locally will not present a problem. However, in other cases, memory might become an issue—particularly if your CA supports an RA and a large number of CRLs end up being stored on your router.

To save NVRAM space, you can specify that certificates and CRLs should not be stored locally, but should be retrieved from the CA when needed. This will save NVRAM space but could result in a slight performance impact.

To specify that certificates and CRLs should not be stored locally on your router, but should be retrieved when required, turn on query mode by using the following command in global configuration mode:

Command	Purpose
`crypto ca certificate query`	Turns on query mode, which causes certificates and CRLs to not be stored locally.

Note Query mode may affect availability if the CA is down.

If you do not turn on query mode at this time, but later decide that you should, you can turn on query mode at that time even if certificates and CRLs have already been stored on your router. In this case, when you turn on query mode, the stored certificates and CRLs will be deleted from the router after you save your configuration. (If you copy your configuration to a TFTP site prior to turning on query mode, you will save any stored certificates and CRLs at the TFTP site.)

If you turn on query mode now, you can turn off query mode later if you wish. If you turn off query mode later, you could also perform the **copy system:running-config nvram:startup-config** command at that time to save all current certificates and CRLs to NVRAM (otherwise they could be lost during a reboot and would need to be retrieved the next time they were needed by your router).

Configuring the Router's Host Name and IP Domain Name

You must configure the router's host name and IP domain name if this has not already been done. This is required because the router assigns a fully qualified domain name (FQDN) to the keys and certificates used by IPSec, and the FQDN is based on the host name and IP domain name you assign to the router. For example, a certificate is named "router20.example.com" based on a router host name of "router20" and a router IP domain name of "example.com."

To configure the router's host name and IP domain name, use the following commands in global configuration mode:

Step	Command	Purpose
1	`hostname` *`name`*	Configures the router's host name.
2	`ip domain-name` *`name`*	Configures the router's IP domain name.

Generating an RSA Key Pair

RSA key pairs are used to sign and encrypt IKE key management messages and are required before you can obtain a certificate for your router.

To generate an RSA key pair, use the following command in global configuration mode:

Command	Purpose
`crypto key generate rsa [usage-keys]`	Generates an RSA key pair. Use the **usage-keys** keyword to specify special usage keys instead of general purpose keys. See the *Cisco IOS Security Command Reference* for an explanation of special usage vs. general purpose keys for this command.

Declaring a Certification Authority

You should declare one certification authority (CA) to be used by your router.

To declare a CA, use the following commands, beginning in global configuration mode:

Step	Command	Purpose
1	`crypto ca identity` *`name`*	Declares a CA. The name should be the CA's domain name. This command puts you into the ca-identity configuration mode.
2	`enrollment url` *`url`*	Specifies the URL of the CA. (The URL should include any non-standard cgi-bin script location.)
3	`enrollment mode ra`	If your CA system provides a Registration Authority (RA), specify RA mode.

Step	Command	Purpose
4	`query url` *url*	If your CA system provides an RA and supports the LDAP protocol, specify the location of the LDAP server.
5	`enrollment retry period` *minutes*	(Optional) Specifies a retry period. After requesting a certificate, the router waits to receive a certificate from the CA. If the router does not receive a certificate within a period of time (the retry period) the router will send another certificate request. You can change the retry period from the default of 1 minute.
6	`enrollment retry count` *number*	(Optional) Specifies how many times the router will continue to send unsuccessful certificate requests before giving up. By default, the router will never give up trying.
7	`crl optional`	(Optional) Specifies that other peers' certificates can still be accepted by your router even if the appropriate CRL is not accessible to your router.
8	`exit`	Exits ca-identity configuration mode.

The trade-off between security and availability is determined by the **query url** and **crl optional** commands, as shown in Table 26.

Table 26 Security and CA Availability

	Query - Yes	Query - No
CRL Optional - Yes	Sessions will go through even if the CA is not available, but the certificate might have been revoked.	Sessions will go through even if the CA is not available, but the certificate might have been revoked.
CRL Optional - No	Certificates will not be accepted if the CA is not available.	Sessions will go through, and will be verified against the CRL stored locally.

Authenticating the CA

The router needs to authenticate the CA. It does this by obtaining the CA's self-signed certificate which contains the CA's public key. Because the CA's certificate is self-signed (the CA signs its own certificate) the CA's public key should be manually authenticated by contacting the CA administrator to compare the CA certificate's fingerprint when you perform this step.

To get the CA's public key, use the following command in global configuration mode:

Command	Purpose
`crypto ca authenticate` *name*	Gets the CA's public key. Use the same *name* that you used when declaring the CA with the **crypto ca identity** command.

Requesting Your Own Certificate(s)

You need to obtain a signed certificate from the CA for each of your router's RSA key pairs. If you generated general purpose RSA keys, your router only has one RSA key pair and needs only one certificate. If you previously generated special usage RSA keys, your router has two RSA key pairs and needs two certificates.

To request signed certificates from the CA, use the following command in global configuration mode:

Command	Purpose
`crypto ca enroll` *name*	Requests certificates for all of your RSA key pairs. This command causes your router to request as many certificates as there are RSA key pairs, so you only need to perform this command once, even if you have special usage RSA key pairs. **Note** This command requires you to create a challenge password that is not saved with the configuration. This password is required in the event your certificate needs to be revoked, so remember this password.

Note If your router reboots after you issued the **crypto ca enroll** command but before you received the certificate(s), you must reissue the command and notify the CA administrator.

Saving Your Configuration

Always remember to save your work when you make configuration changes.

Use the **copy system:running-config nvram:startup-config** command to save your configuration—this command includes saving RSA keys to private NVRAM. RSA keys are *not* saved with your configuration when you use a **copy system:running-config rcp:** or **copy system:running-config tftp:** command.

Monitoring and Maintaining Certification Authority Interoperability

The following tasks are optional, depending on your particular requirements:

- Requesting a Certificate Revocation List
- Deleting Your Router's RSA Keys
- Deleting Peer's Public Keys
- Deleting Certificates from the Configuration
- Viewing Keys and Certificates

Requesting a Certificate Revocation List

You can request a certificate revocation list (CRL) only if your CA does not support a Registration Authority (RA). The following description and task applies only when the CA does not support an RA.

When your router receives a certificate from a peer, your router will download a CRL from the CA. Your router then checks the CRL to make sure the certificate the peer sent has not been revoked. (If the certificate appears on the CRL, your router will not accept the certificate and will not authenticate the peer.)

A CRL can be reused with subsequent certificates until the CRL expires if query mode is off. If your router receives a peer's certificate after the applicable CRL has expired, the router will download the new CRL.

If your router has a CRL which has not yet expired, but you suspect that the CRL's contents are out of date, you can request that the latest CRL be immediately downloaded to replace the old CRL.

To request immediate download of the latest CRL, use the following command in global configuration mode:

Command	Purpose
`crypto ca crl request` *name*	Requests an updated CRL. This command replaces the currently stored CRL at your router with the newest version of the CRL.

Deleting Your Router's RSA Keys

There might be circumstances where you would want to delete your router's RSA keys. For example, if you believe the RSA keys were compromised in some way and should no longer be used, you should delete the keys.

To delete all of your router's RSA keys, use the following command in global configuration mode:

Command	Purpose
`crypto key zeroize rsa`	Deletes all of your router's RSA keys.

After you delete a router's RSA keys, you should also complete these two additional tasks:

- Ask the CA administrator to revoke your router's certificates at the CA; you must supply the challenge password you created when you originally obtained the router's certificates with the **crypto ca enroll** command.
- Manually remove the router's certificates from the router configuration as described in the "Deleting Certificates from the Configuration" section.

Deleting Peer's Public Keys

There might be circumstances where you would want to delete other peer's RSA public keys from your router's configuration. For example, if you no longer trust the integrity of a peer's public key, you should delete the key.

To delete a peer's RSA public key, use the following commands, beginning in global configuration mode:

Step	Command	Purpose
1	`crypto key pubkey-chain rsa`	Enters public key configuration mode.
2	`no named-key` *key-name* `[encryption	signature]` or `no addressed-key` *key-address* `[encryption	signature]`	Deletes a remote peer's RSA public key. Specify the peer's fully qualified domain name or the remote peer's IP address.
3	`exit`	Returns to global configuration mode.

Deleting Certificates from the Configuration

If the need arises, you can delete certificates that are saved at your router. Your router saves its own certificate(s), the CA's certificate, and any RA certificates (unless you put the router into query mode per the "Managing NVRAM Memory Usage" section).

To delete your router's certificate or RA certificates from your router's configuration, use the following commands in global configuration mode:

Step	Command	Purpose
1	`show crypto ca certificates`	Displays the certificates stored on your router; note (or copy) the serial number of the certificate you wish to delete.
2	`crypto ca certificate chain` *`name`*	Enters certificate chain configuration mode.
3	`no certificate` *`certificate-serial-number`*	Deletes the certificate.

To delete the CA's certificate, you must remove the entire CA identity, which also removes all certificates associated with the CA—your router's certificate, the CA certificate, and any RA certificates.

To remove a CA identity, use the following command in global configuration mode:

Command	Purpose
`no crypto ca identity` *`name`*	Deletes all identity information and certificates associated with the CA.

Viewing Keys and Certificates

To view keys and certificates, use the following commands in EXEC mode:

Step	Command	Purpose
1	`show crypto key mypubkey rsa`	Displays your router's RSA public keys.
2	`show crypto key pubkey-chain rsa`	Displays a list of all the RSA public keys stored on your router. These include the public keys of peers who have sent your router their certificates during peer authentication for IPSec.
3	`show crypto key pubkey-chain rsa [name` *`key-name`* `| address` *`key-address`*`]`	Displays details of a particular RSA public key stored on your router.
4	`show crypto ca certificates`	Displays information about your certificate, the CA's certificate, and any RA certificates.

What to Do Next

After you are done configuring this feature, you should configure IKE and IPSec. IKE configuration is described in the "Configuring Internet Key Exchange Security Protocol" chapter. IPSec configuration is described in the "Configuring IPSec Network Security" chapter.

CA Interoperability Configuration Examples

The following configuration is for a router named "myrouter." In this example IPSec is configured and the IKE protocol and CA interoperability are configured in support of IPSec.

In this example general purpose RSA keys were generated, but you will notice that the keys are not saved or displayed in the configuration.

Comments are included within the configuration to explain various commands.

```
!
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
! CA interoperability requires you to configure your router's hostname:
hostname myrouter
!
enable secret 5 <removed>
enable password <removed>
!
! CA interoperability requires you to configure your router's IP domain name:
ip domain-name example.com
ip name-server 172.29.2.132
ip name-server 192.168.30.32
!
! The following configures a transform set (part of IPSec configuration):
crypto ipsec transform-set my-transformset esp-3des esp-sha-hmac
!
! The following declares the CA. (In this example, the CA does not support an RA.)
crypto ca identity example.com
 enrollment url http://ca_server
!
! The following shows the certificates and CRLs stored at the router, including
!   the CA certificate (shown first), the router's certificate (shown next)
!   and a CRL (shown last).
crypto ca certificate chain example.com
! The following is the CA certificate
!   received via the 'crypto ca authenticate' command:
 certificate ca 3051DF7169BEE31B821DFE4B3A338E5F
  30820182 3082012C A0030201 02021030 51DF7169 BEE31B82 1DFE4B3A 338E5F30
  0D06092A 864886F7 0D010104 05003042 31163014 06035504 0A130D43 6973636F
  20537973 74656D73 3110300E 06035504 0B130744 65767465 73743116 30140603
  55040313 0D434953 434F4341 2D554C54 5241301E 170D3937 31323032 30313036
  32385A17 0D393831 32303230 31303632 385A3042 31163014 06035504 0A130D43
  6973636F 20537973 74656D73 3110300E 06035504 0B130744 65767465 73743116
  30140603 55040313 0D434953 434F4341 2D554C54 5241305C 300D0609 2A864886
  F70D0101 01050003 4B003048 024100C1 B69D7BF6 34E4EE28 A84E0DC6 FCA4DEA8
  04D89E50 C5EBE862 39D51890 D0D4B732 678BDBF2 80801430 E5E56E7C C126E2DD
  DBE9695A DF8E5BA7 E67BAE87 29375302 03010001 300D0609 2A864886 F70D0101
  04050003 410035AA 82B5A406 32489413 A7FF9A9A E349E5B4 74615E05 058BA3CE
  7C5F00B4 019552A5 E892D2A3 86763A1F 2852297F C68EECE1 F41E9A7B 2F38D02A
  B1D2F817 3F7B
  quit
! The following is the router's certificate
!   received via the 'crypto ca enroll' command:
certificate 7D28D4659D22C49134B3D1A0C2C9C8FC
  308201A6 30820150 A0030201 0202107D 28D4659D 22C49134 B3D1A0C2 C9C8FC30
  0D06092A 864886F7 0D010104 05003042 31163014 06035504 0A130D43 6973636F
  20537973 74656D73 3110300E 06035504 0B130744 65767465 73743116 30140603
  55040313 0D434953 434F4341 2D554C54 5241301E 170D3938 30343234 30303030
  30305A17 0D393930 34323432 33353935 395A302F 311D301B 06092A86 4886F70D
  01090216 0E73636F 742E6369 73636F2E 636F6D31 0E300C06 03550405 13053137
  41464230 5C300D06 092A8648 86F70D01 01010500 034B0030 48024100 A207ED75
  DE8A9BC4 980958B7 28ADF562 1371D043 1FC93C24 8E9F8384 4D1A2407 60CBD7EC
  B15BD782 A687CA49 883369BE B35A4219 8FE742B0 91CF76EE 07EC9E69 02030100
  01A33530 33300B06 03551D0F 04040302 05A03019 0603551D 11041230 10820E73
  636F742E 63697363 6F2E636F 6D300906 03551D13 04023000 300D0609 2A864886
  F70D0101 04050003 410085F8 A5AFA907 B38731A5 0195D921 D8C45EFD B6082C28
  04A88CEC E9EC6927 F24874E4 30C4D7E2 2686E0B5 77F197E4 F82A8BA2 1E03944D
  286B661F 0305DF5F 3CE7
  quit
! The following is a CRL received by the router (via the router's own action):
crl
  3081C530 71300D06 092A8648 86F70D01 01020500 30423116 30140603 55040A13
  0D436973 636F2053 79737465 6D733110 300E0603 55040B13 07446576 74657374
  31163014 06035504 03130D43 4953434F 43412D55 4C545241 170D3938 30333233
  32333232 31305A17 0D393930 34323230 30303030 305A300D 06092A86 4886F70D
  01010205 00034100 7AA83057 AC5E5C65 B9812549 37F11B7B 5CA4CAED 830B3955
  A4DDD268 F567E29A E4B34691 C2162BD1 0540D7E6 5D6650D1 81DBBF1D 788F1DAC
  BBF761B2 81FCC0F1
  quit
!
! The following is an IPSec crypto map (part of IPSec configuration):
crypto map map-to-remotesite 10 ipsec-isakmp
 set peer 172.21.114.196
 set transform-set my-transformset
 match address 124
!
!
interface Loopback0
 ip address 10.0.0.1 255.0.0.0
!
interface Tunnel0
 ip address 10.0.0.2 255.0.0.0
 ip mtu 1490
 no ip route-cache
 no ip mroute-cache
 tunnel source 10.10.0.1
 tunnel destination 172.21.115.119
!
interface FastEthernet0/0
 ip address 172.21.115.118 255.255.255.240
 no ip mroute-cache
 loopback
 no keepalive
 shutdown
 media-type MII
 full-duplex
!
! The IPSec crypto map is applied to interface Ethernet1/0:
interface Ethernet1/0
 ip address 172.21.114.197 255.255.255.0
 bandwidth 128
 no keepalive
 no fair-queue
 no cdp enable
 crypto map map-to-remotesite
!
crypto isakmp policy 15
 encryption 3des
 hash md5
 authentication rsa-sig
 group 2
 lifetime 5000
crypto isakmp policy 20
 authentication pre-share
 lifetime 10000
crypto isakmp key 1234567890 address 171.69.224.33
```

Configuring Internet Key Exchange Security Protocol

This chapter describes how to configure the Internet Key Exchange (IKE) protocol. IKE is a key management protocol standard that is used in conjunction with the IPSec standard. IPSec is an IP security feature that provides robust authentication and encryption of IP packets.

IPSec can be configured without IKE, but IKE enhances IPSec by providing additional features, flexibility, and ease of configuration for the IPSec standard.

IKE is a hybrid protocol which implements the Oakley key exchange and Skeme key exchange inside the Internet Security Association and Key Management Protocol (ISAKMP) framework. (ISAKMP, Oakley, and Skeme are security protocols implemented by IKE.)

For a complete description of the IKE commands used in this chapter, refer to the "Internet Key Exchange Security Protocol Commands" chapter in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

In This Chapter

This chapter includes the following sections:

- IKE Overview
- IKE Configuration Task List
- What to Do Next
- IKE Configuration Examples

IKE Overview

IKE automatically negotiates IPSec security associations (SAs) and enables IPSec secure communications without costly manual preconfiguration. Specifically, IKE provides these benefits:

- Eliminates the need to manually specify all the IPSec security parameters in the crypto maps at both peers.
- Allows you to specify a lifetime for the IPSec security association.
- Allows encryption keys to change during IPSec sessions.
- Allows IPSec to provide anti-replay services.
- Permits certification authority (CA) support for a manageable, scalable IPSec implementation.
- Allows dynamic authentication of peers.

This section includes the following sections:

- Supported Standards
- List of Terms
- IKE Aggressive Mode Behavior

Supported Standards

Cisco implements the following standards:

- **IPSec**—IP Security Protocol. IPSec is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer; it uses IKE to handle negotiation of protocols and algorithms based on local policy, and to generate the encryption and authentication keys to be used by IPSec. IPSec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

  For more information on IPSec, see the "Configuring IPSec Network Security" chapter.

- **Internet Key Exchange (IKE)**—A hybrid protocol which implements Oakley and Skeme key exchanges inside the ISAKMP framework. While IKE can be used with other protocols, its initial implementation is with the IPSec protocol. IKE provides authentication of the IPSec peers, negotiates IPSec keys, and negotiates IPSec security associations.

  IKE is implemented per RFC 2409, "The Internet Key Exchange."

- **ISAKMP**—The Internet Security Association and Key Management Protocol. A protocol framework which defines payload formats, the mechanics of implementing a key exchange protocol, and the negotiation of a security association.

  ISAKMP is implemented per the latest version of the "Internet Security Association and Key Management Protocol (ISAKMP)" Internet Draft (RFC 2408).

- **Oakley**—A key exchange protocol which defines how to derive authenticated keying material.
- **Skeme**—A key exchange protocol which defines how to derive authenticated keying material, with rapid key refreshment.

The component technologies implemented for use by IKE include:

- **DES**—The Data Encryption Standard (DES) is used to encrypt packet data. IKE implements the 56-bit DES-CBC with Explicit IV standard. Cipher Block Chaining (CBC) requires an initialization vector (IV) to start encryption. The IV is explicitly given in the IPSec packet.

  Cisco IOS also implements Triple DES (168-bit) encryption, depending on the software versions available for a specific platform. Triple DES (3DES) is a strong form of encryption that allows sensitive information to be transmitted over untrusted networks. It enables customers, particularly in the finance industry, to utilize network layer encryption.

Caution Cisco IOS images with strong encryption (including, but not limited to, 56-bit data encryption feature sets) are subject to United States government export controls, and have a limited distribution. Images to be installed outside the United States require an export license. Customer orders might be denied or subject to delay due to United States government regulations. Contact your sales representative or distributor for more information, or send e-mail to export@cisco.com.

- **Diffie-Hellman**—A public-key cryptography protocol which allows two parties to establish a shared secret over an unsecure communications channel. Diffie-Hellman is used within IKE to establish session keys. 768-bit and 1024-bit Diffie-Hellman groups are supported.
- **MD5 (HMAC variant)**—MD5 (Message Digest 5) is a hash algorithm used to authenticate packet data. HMAC is a variant which provides an additional level of hashing.
- **SHA (HMAC variant)**—SHA (Secure Hash Algorithm) is a hash algorithm used to authenticate packet data. HMAC is a variant which provides an additional level of hashing.
- **RSA signatures** and **RSA encrypted nonces**—RSA is the public key cryptographic system developed by Ron Rivest, Adi Shamir, and Leonard Adleman. RSA signatures provides non-repudiation while RSA encrypted nonces provide repudiation.

IKE interoperates with the following standard:

X.509v3 certificates—Used with the IKE protocol when authentication requires public keys. This certificate support allows the protected network to scale by providing the equivalent of a digital ID card to each device. When two devices wish to communicate, they exchange digital certificates to prove their identity (thus removing the need to manually exchange public keys with each peer or to manually specify a shared key at each peer).

List of Terms

anti-replay—A security service in which the receiver can reject old or duplicate packets in order to protect itself against replay attacks. IPSec provides optional anti-replay services by use of a sequence number combined with the use of authentication.

data authentication—Includes two concepts:

- Data integrity (verify that data has not been altered).
- Data origin authentication (verify that the data was actually sent by the claimed sender).

Data authentication can refer either to integrity alone or to both of these concepts (although data origin authentication is dependent upon data integrity).

peer—In the context of this chapter, a peer refers to a router or other device that participates in IPSec and IKE.

perfect forward secrecy (PFS)—A cryptographic characteristic associated with a derived shared secret value. With PFS, if one key is compromised, previous and subsequent keys are not also compromised, because subsequent keys are not derived from previous keys.

repudiation—A quality that prevents a third party from being able to prove that a communication between two other parties ever took place. This is a desirable quality if you do not want your communications to be traceable. **Non-repudiation** is the opposite quality—a third party can prove that a communication between two other parties took place. Non-repudiation is desirable if you want to be able to trace your communications and prove that they occurred.

security association—A security association (SA) describes how two or more entities will utilize security services to communicate securely. For example, an IPSec SA defines the encryption algorithm (if used), the authentication algorithm, and the shared session key to be used during the IPSec connection.

Both IPSec and IKE require and use SAs to identify the parameters of their connections. IKE can negotiate and establish its own SA. The IPSec SA is established either by IKE or by manual user configuration.

IKE Aggressive Mode Behavior

This section describes IKE aggressive mode behavior when using Cisco IOS software.

IKE has two phases of key negotiation: phase 1 and phase 2. Phase 1 negotiates a security association (a key) between two IKE peers. The key negotiated in phase 1 enables IKE peers to communicate securely in phase 2. During phase 2 negotiation, IKE establishes keys (security associations) for other applications, such as IPSec.

Phase 1 negotiation can occur using one of two modes: main mode or aggressive mode. Main mode tries to protect all information during the negotiation, meaning that no information is available to a potential attacker. When using main mode, the identities of the two sides are hidden. While this mode of operation is very secure, it is more costly in terms of the time it takes to complete the negotiation. Aggressive mode takes less time to negotiate keys between peers; however, it gives up some of the security provided by main mode negotiation. For example, the identities of the two parties trying to establish a security association are exposed to an eavesdropper.

The two modes serve different purposes and have different strengths. Main mode is slower than aggressive mode, but main mode is more secure and more flexible because if can offer an IKE peer more security proposals than aggressive mode. Aggressive mode is less flexible and not as secure, but much faster.

In Cisco IOS, the two modes are not configurable. The default action for IKE authentication (rsa-sig, rsa-encr, or pre-shared) is to initiate main mode; however, in cases where there is no corresponding information to initiate authentication, and there is a pre-shared key associated with the hostname of the peer, Cisco IOS can initiate aggressive mode. Cisco IOS will respond in aggressive mode to an IKE peer that initiates aggressive mode.

IKE Configuration Task List

To configure IKE, perform the tasks in the following sections. The tasks in the first three sections are required; the remaining may be optional, depending on what parameters are configured.

- Enabling or Disabling IKE (Required)
- Ensuring Access Lists Are Compatible with IKE (Required)
- Creating IKE Policies (Required)
- Manually Configuring RSA Keys (Optional, depending on IKE parameters)
- Configuring Pre-Shared Keys (Optional, depending on IKE parameters)
- Configuring Internet Key Exchange Mode Configuration (Optional)
- Configuring Tunnel Endpoint Discovery (Optional)
- Clearing IKE Connections (Optional)
- Troubleshooting IKE (Optional)

For IKE configuration examples, refer to the "IKE Configuration Examples" section at the end of this chapter.

Enabling or Disabling IKE

IKE is enabled by default. IKE does not have to be enabled for individual interfaces, but is enabled globally for all interfaces at the router.

If you do not want IKE to be used with your IPSec implementation, you can disable it at all IPSec peers.

If you disable IKE, you will have to make these concessions at the peers:

- You must manually specify all the IPSec security associations in the crypto maps at all peers. (Crypto map configuration is described in the "Configuring IPSec Network Security" chapter.)
- The peers' IPSec security associations will never time out for a given IPSec session.
- During IPSec sessions between the peers, the encryption keys will never change.
- Anti-replay services will not be available between the peers.
- Certification authority (CA) support cannot be used.

To disable or enable IKE, use one of the following commands in global configuration mode:

Command	Purpose
`no crypto isakmp enable`	Disables IKE.
`crypto isakmp enable`	Enables IKE.

If you disable IKE, you can skip the rest of the tasks in this chapter and go directly to IPSec configuration as described in the "Configuring IPSec Network Security" chapter.

Ensuring Access Lists Are Compatible with IKE

IKE negotiation uses UDP on port 500. Ensure that your access lists are configured so that UDP port 500 traffic is not blocked at interfaces used by IKE and IPSec. In some cases you might need to add a statement to your access lists to explicitly permit UDP port 500 traffic.

Creating IKE Policies

You must create IKE policies at each peer. An IKE policy defines a combination of security parameters to be used during the IKE negotiation.

To create an IKE policy, follow the guidelines in these sections:

- Why Do You Need to Create These Policies?
- What Parameters Do You Define in a Policy?
- How Do IKE Peers Agree upon a Matching Policy?
- Which Value Should You Select for Each Parameter?
- Creating Policies
- Additional Configuration Required for IKE Policies

Why Do You Need to Create These Policies?

IKE negotiations must be protected, so each IKE negotiation begins by each peer agreeing on a common (shared) IKE policy. This policy states which security parameters will be used to protect subsequent IKE negotiations and mandates how the peers are authenticated.

After the two peers agree upon a policy, the security parameters of the policy are identified by a security association established at each peer, and these security associations apply to all subsequent IKE traffic during the negotiation.

You can create multiple, prioritized policies at each peer to ensure that at least one policy will match a remote peer's policy.

What Parameters Do You Define in a Policy?

There are five parameters to define in each IKE policy:

Parameter	Accepted Values	Keyword	Default Value
encryption algorithm	56-bit DES-CBC	**des**	56-bit DES-CBC
	168-bit DES	**3des**	168-bit DES
hash algorithm	SHA-1 (HMAC variant)	**sha**	SHA-1
	MD5 (HMAC variant)	**md5**	
authentication method	RSA signatures	**rsa-sig**	RSA signatures
	RSA encrypted nonces	**rsa-encr**	
	pre-shared keys	**pre-share**	
Diffie-Hellman group identifier	768-bit Diffie-Hellman or	**1**	768-bit Diffie-Hellman
	1024-bit Diffie-Hellman	**2**	
security association's lifetime[1]	Can specify any number of seconds	—	86400 seconds (one day)

[1] For information about this lifetime and how it is used, see the command description for the **lifetime (IKE policy)** command.

These parameters apply to the IKE negotiations when the IKE security association is established.

How Do IKE Peers Agree upon a Matching Policy?

When the IKE negotiation begins, IKE looks for an IKE policy that is the same on both peers. The peer that initiates the negotiation will send all its policies to the remote peer, and the remote peer will try to find a match. The remote peer looks for a match by comparing its own highest priority policy against the other peer's received policies. The remote peer checks each of its policies in order of its priority (highest priority first) until a match is found.

A match is made when both policies from the two peers contain the same encryption, hash, authentication, and Diffie-Hellman parameter values, and when the remote peer's policy specifies a lifetime less than or equal to the lifetime in the policy being compared. (If the lifetimes are not identical, the shorter lifetime—from the remote peer's policy—will be used.)

If no acceptable match is found, IKE refuses negotiation and IPSec will not be established.

If a match is found, IKE will complete negotiation, and IPSec security associations will be created.

Note Depending on which authentication method is specified in a policy, additional configuration might be required (as described in the "Additional Configuration Required for IKE Policies" section). If a peer's policy does not have the required companion configuration, the peer will not submit the policy when attempting to find a matching policy with the remote peer.

Which Value Should You Select for Each Parameter?

You can select certain values for each parameter, per the IKE standard. But why chose one value over another?

If you are interoperating with a device that supports only one of the values for a parameter, your choice is limited to the other device's supported value. Aside from this, there is often a trade-off between security and performance, and many of these parameter values represent such a trade-off. You should evaluate the level of your network's security risks and your tolerance for these risks. Then the following tips might help you select which value to specify for each parameter:

- The encryption algorithm has two options: 56-bit DES-CBC and 168-bit DES.
- The hash algorithm has two options: SHA-1 and MD5.

  MD5 has a smaller digest and is considered to be slightly faster than SHA-1. There has been a demonstrated successful (but extremely difficult) attack against MD5; however, the HMAC variant used by IKE prevents this attack.
- The authentication method has three options: RSA signatures, RSA encrypted nonces, and pre-shared keys.
  - RSA signatures provides non-repudiation for the IKE negotiation (you can prove to a third party after the fact that you did indeed have an IKE negotiation with the remote peer).

    RSA signatures requires use of a certification authority (CA). Using a CA can dramatically improve the manageability and scalability of your IPSec network. Additionally, RSA signature-based authentication uses only two public key operations while RAS encryption uses four public key operations, which is costlier in terms of overall performance.
  - RSA encrypted nonces provides repudiation for the IKE negotiation (you cannot prove to a third party that you had an IKE negotiation with the remote peer). This is used to prevent a

    RSA encrypted nonces require that peers possess each other's public keys but do not use a certification authority. Instead, there are two ways for peers to get each others' public keys:

    1) During configuration you manually configure RSA keys (as described in the "Manually Configuring RSA Keys" section).

    2) If your local peer has previously used RSA signatures during a successful IKE negotiation with a remote peer, your local peer already possesses the remote peer's public key. (The peers' public keys are exchanged during the RSA-signatures-based IKE negotiations.)
  - Pre-shared keys are clumsy to use if your secured network is large, and do not scale well with a growing network. However, they do not require use of a certification authority, as do RSA signatures, and might be easier to set up in a small network with fewer than 10 nodes. RSA signatures also can be considered more secure when compared to pre-shared key authentication.
- The Diffie-Hellman group identifier has two options: 768-bit or 1024-bit Diffie-Hellman.

  The 1024-bit Diffie-Hellman option is harder to crack, but requires more CPU time to execute.
- The security association's lifetime can be set to any value.

As a general rule, the shorter the lifetime (up to a point), the more secure your IKE negotiations will be. However, with longer lifetimes, future IPSec security associations can be set up more quickly. For more information about this parameter and how it is used, see the command description for the **lifetime (IKE policy)** command.

Creating Policies

You can create multiple IKE policies, each with a different combination of parameter values. For each policy that you create, you assign a unique priority (1 through 10,000, with 1 being the highest priority).

You can configure multiple policies on each peer—but at least one of these policies must contain exactly the same encryption, hash, authentication, and Diffie-Hellman parameter values as one of the policies on the remote peer. (The lifetime parameter does not necessarily have to be the same; see details in the "How Do IKE Peers Agree upon a Matching Policy?" section.)

If you do not configure any policies, your router will use the default policy, which is always set to the lowest priority, and which contains each parameter's default value.

To configure a policy, use the following commands, beginning in global configuration mode:

Step	Command	Purpose
1	`crypto isakmp policy` *`priority`*	Identifies the policy to create. (Each policy is uniquely identified by the priority number you assign.) (This command puts you into the config-isakmp command mode.)
2	`encryption {des | 3des}`	Specifies the encryption algorithm.
3	`hash {sha | md5}`	Specifies the hash algorithm.
4	`authentication {rsa-sig | rsa-encr | pre-share}`	Specifies the authentication method.
5	`group {1 | 2}`	Specifies the Diffie-Hellman group identifier.
6	`lifetime` *`seconds`*	Specifies the security association's lifetime.
7	`exit`	Exits the config-isakmp command mode.
8	`exit`	Exits the global configuration mode.
9	`show crypto isakmp policy`	(Optional) Displays all existing IKE policies. (Use this command in EXEC mode.)

If you do not specify a value for a parameter, the default value is assigned.

Note The default policy and the default values for configured policies do not show up in the configuration when you issue a **show running** command. Instead, to see the default policy and any default values within configured policies, use the **show crypto isakmp policy** command.

Additional Configuration Required for IKE Policies

Depending on which authentication method you specify in your IKE policies, you need to do certain additional configuration before IKE and IPSec can successfully use the IKE policies.

Each authentication method requires additional companion configuration as follows:

- RSA signatures method:

  If you specify RSA signatures as the authentication method in a policy, you must configure the peers to obtain certificates from a certification authority (CA). (And, of course, the CA must be properly configured to issue the certificates.) Configure this certificate support as described in the "Configuring Certification Authority Interoperability" chapter.

  The certificates are used by each peer to securely exchange public keys. (RSA signatures requires that each peer has the remote peer's public signature key.) When both peers have valid certificates, they will automatically exchange public keys with each other as part of any IKE negotiation in which RSA signatures are used.

- RSA encrypted nonces method:

  If you specify RSA encrypted nonces as the authentication method in a policy, you need to ensure that each peer has the other peers' public keys.

  Unlike RSA signatures, the RSA encrypted nonces method does not use certificates to exchange public keys. Instead, you ensure that each peer has the others' public keys by either:

  — Manually configuring RSA keys as described in the "Manually Configuring RSA Keys" section.

    or

  — Ensuring that an IKE exchange using RSA signatures has already occurred between the peers. (The peers' public keys are exchanged during the RSA-signatures-based IKE negotiations.)

    To make this happen, specify two policies: a higher-priority policy with RSA encrypted nonces, and a lower-priority policy with RSA signatures. When IKE negotiations occur, RSA signatures will be used the first time because the peers do not yet have each others' public keys. Then, future IKE negotiations will be able to use RSA encrypted nonces because the public keys will have been exchanged.

    Of course, this alternative requires that you have certification authority support configured.

- Pre-shared keys authentication method:

  If you specify pre-shared keys as the authentication method in a policy, you must configure these pre-shared keys as described in the "Configuring Pre-Shared Keys" section.

If RSA encryption is configured and signature mode is negotiated, the peer will request both signature and encryption keys. Basically, the router will request as many keys as the configuration will support. If RSA encryption is not configured, it will just request a signature key.

Manually Configuring RSA Keys

Manually configure RSA keys when you specify RSA encrypted nonces as the authentication method in an IKE policy and you are not using a certification authority (CA).

To manually configure RSA keys, perform these tasks at each IPSec peer that uses RSA encrypted nonces in an IKE policy:

- Generating RSA Keys
- Setting ISAKMP Identity
- Specifying All the Other Peers' RSA Public Keys

Generating RSA Keys

To generate RSA keys, use the following commands starting in global configuration mode:

Step	Command	Purpose
1	`crypto key generate rsa [usage-keys]`	Generates RSA keys.
2	`show crypto key mypubkey rsa`	Displays the generated RSA public key (in EXEC mode).

Remember to repeat these tasks at each peer (without CA support) that uses RSA encrypted nonces in an IKE policy.

Setting ISAKMP Identity

You should set the ISAKMP identity for each peer that uses pre-shared keys in an IKE policy.

When two peers use IKE to establish IPSec security associations, each peer sends its identity to the remote peer. Each peer sends either its host name or its IP address, depending on how you have the router's ISAKMP identity set.

By default, a peer's ISAKMP identity is the peer's IP address. If appropriate, you could change the identity to be the peer's host name instead. As a general rule, set all peers' identities the same way—either all peers should use their IP address, or all peers should use their host name. If some peers use their host name and some peers use their IP address to identify themselves to each other, IKE negotiations could fail if a remote peer's identity is not recognized and a DNS lookup is unable to resolve the identity.

To set a peer's ISAKMP identity, use the following commands in global configuration mode:

Step	Command	Purpose
1	`crypto isakmp identity {address	hostname}`	**At the local peer**: Specifies the peer's ISAKMP identity by IP address or by host name.[1]
2	`ip host` *hostname address1* `[`*address2...address8*`]`	**At all remote peers**: If the local peer's ISAKMP identity was specified using a host name, maps the peer's host name to its IP address(es) at all the remote peers. (This step might be unnecessary if the host name/address is already mapped in a DNS server.)

[1] See the **crypto isakmp identity** command description for guidelines for when to use the IP address vs. the host name.

Remember to repeat these tasks at each peer that uses pre-shared keys in an IKE policy.

Specifying All the Other Peers' RSA Public Keys

At each peer, specify all the other peers' RSA public keys by using the following commands starting in global configuration mode:

Step	Command	Purpose
1	`crypto key pubkey-chain rsa`	Enters public key configuration mode.
2	`named-key key-name [encryption	signature]` or `addressed-key` *key-address* `[encryption	signature]`	Indicates which remote peer's RSA public key you are going to specify. If the remote peer uses its host name as its ISAKMP identity, use the **named-key** command and specify the remote peer's fully qualified domain name (such as somerouter.example.com) as the *key-name*. If the remote peer uses its IP address as its ISAKMP identity, use the **addressed-key** command and specify the remote peer's IP address as the *key-address*.

Step	Command	Purpose
3	`address ip-address`	If you used a fully qualified domain name to name the remote peer in Step 2 (using the **named-key** command), you can optionally specify the remote peer's IP address.

4	`key-string`
	`key-string`
	`quit`
		Specifies the remote peer's RSA public key. This is the key viewed by the remote peer's administrator previously when he generated his router's RSA keys.

5	—	Repeat Steps 2 through 4 to specify the RSA public keys of all the other IPSec peers that use RSA encrypted nonces in an IKE policy.
6	`exit`	Returns to global configuration mode.

Remember to repeat these tasks at each peer that uses RSA encrypted nonces in an IKE policy.

To view RSA public keys while or after you configure them, use the following command in EXEC mode:

Command	Purpose
`show crypto key pubkey-chain rsa {name key-name	address key-address}`	Displays a list of all the RSA public keys stored on your router, or displays details of a particular RSA public key stored on your router.

Configuring Pre-Shared Keys

To configure pre-shared keys, perform these tasks at each peer that uses pre-shared keys in an IKE policy:

- First, set each peer's ISAKMP identity. Each peer's identity should be set to either its host name or by its IP address. By default, a peer's identity is set to its IP address. Setting ISAKMP identities is described previously in the "Setting ISAKMP Identity" section.
- Next, specify the shared keys at each peer. Note that a given pre-shared key is shared between two peers. At a given peer you could specify the same key to share with multiple remote peers; however, a more secure approach is to specify different keys to share between different pairs of peers.

To specify pre-shared keys at a peer, use the following commands in global configuration mode:

Step	Command	Purpose

1	`crypto isakmp key keystring address peer-address`
	or
	`crypto isakmp key keystring hostname peer-hostname`
		At the local peer: Specifies the shared key to be used with a particular remote peer.
		If the remote peer specified their ISAKMP identity with an address, use the **address** keyword in this step; otherwise use the **hostname** keyword in this step.

2	`crypto isakmp key keystring address peer-address`
	or
	`crypto isakmp key keystring hostname peer-hostname`
		At the remote peer: Specifies the shared key to be used with the local peer. This is the same key you just specified at the local peer.
		If the local peer specified their ISAKMP identity with an address, use the **address** keyword in this step; otherwise use the **hostname** keyword in this step.

3	—	Repeat the previous two steps for each remote peer.

Remember to repeat these tasks at each peer that uses pre-shared keys in an IKE policy.

Configuring Internet Key Exchange Mode Configuration

Internet Key Exchange (IKE) Mode Configuration, as defined by the Internet Engineering Task Force (IETF), allows a gateway to download an IP address (and other network level configuration) to the client as part of an IKE negotiation. Using this exchange, the gateway gives IP addresses to the IKE client to be used as an "inner" IP address encapsulated under IPSec. This provides a known IP address for the client which can be matched against Internet Protocol Security (IPSec) policy.

To implement IPSec Virtual Private Networks (VPNs) between remote access clients with dynamic IP addresses and a corporate gateway, you have to dynamically administer scalable IPSec policy on the gateway once each client is authenticated. With IKE Mode Configuration, the gateway can set up scalable policy for a very large set of clients irrespective of the IP addresses of those clients.

There are two types of IKE Mode Configuration:

- Gateway initiation—Gateway initiates the configuration mode with the client. Once the client responds, the IKE modifies the sender's identity, the message is processed, and the client receives a response.
- Client initiation—Client initiates the configuration mode with the gateway. The gateway responds with an IP address it has allocated for the client.

IKE Mode Configuration has the following restrictions:

- Interfaces with crypto maps which are configured for IKE Mode Configuration may experience a slightly longer connection set up time. This is true even for IKE peers that refuse to be configured or do not respond to the configuration mode request. In both cases, the gateway initiates the configuration of the client.
- At the time of this publication, this feature is an IETF draft with limited support. Therefore, this feature was not designed to enable the configuration mode for every IKE connection by default. Configure this feature at the global crypto map level.
- The following items in the IETF draft are not currently supported:
  - — Configuration attributes other than INTERNAL_IP_ADDRESS
  - — Unprotected exchanges

There are two steps to configuring IKE Mode Configuration on a router:

1. Define the pool of IP addresses.
2. Define which crypto maps should attempt to configure clients.

To configure IKE Mode Configuration on your Cisco access router, use the following commands in global configuration mode:

Step	Command	Purpose
1	`router(config)# ip local pool pool-name start-addr end-addr`	Existing local address pools are used to define a set of addresses. To define a local address pool, use the existing **ip local pool** command. For more information on the **ip local pool** command, refer to the *Cisco IOS Dial Services Command Reference.*
2	`router(config)# crypto isakmp client configuration address-pool local pool-name`	The local pool references the IKE configuration. To reference this local address pool in the IKE configuration, use the new **crypto isakmp client configuration address-pool local** command. For more information on the **crypto isakmp client configuration address-pool local** command, refer to the *Cisco IOS Security Command Reference.*
3	`router(config)# crypto map tag client configuration address [initiate | respond]`	To configure IKE Mode Configuration in global crypto map configuration mode, use the new **crypto map client configuration address** command. For more information on the **crypto map client configuration address** command, refer to the *Cisco IOS Security Command Reference.*

Configuring Tunnel Endpoint Discovery

Tunnel Endpoint Discovery is an enhancement to the IP Security Protocol (IPSec) feature. Defining a dynamic crypto map allows you to be able to dynamically determine an IPSec peer; however, only the receiving router has this ability. With Tunnel Endpoint Discovery, the initiating router can dynamically determine an IPSec peer for secure IPSec communications.

Dynamic Tunnel Endpoint Discovery allows IPSec to scale to large networks by reducing multiple encryptions, reducing the setup time, and allowing for simple configurations on participating peer routers. Each node has a simple configuration that defines the local network that the router is protecting and the IPSec transforms that are required.

Tunnel Endpoint Discovery has the following restrictions:

- The Internet Key Exchange (IKE) cannot occur until the peer is identified. You can identify the peer using the **crypto dynamic-map** command. Using the **discover** keyword with this command, the receiving router sends a probe out to receive a response from the peer router.
- This feature is only available on dynamic crypto maps. The dynamic crypto map template is based on the dynamic crypto map performing peer discovery. Although there are no access-list restrictions on the dynamic crypto map template, the dynamic crypto map template should cover data sourced from the protected traffic and the receiving router using the **any** keyword. When using the **any** keyword, include explicit **deny** statements to exempt routing protocol traffic prior to entering the **permit any** command.

To create a dynamic crypto map entry with Tunnel Endpoint Discovery configured, use the following commands, beginning in crypto-map configuration mode:

Step	Command	Purpose
1	**crypto dynamic-map** *dynamic-map-name dynamic-map-number* **set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*] **match address** *access-list-id* **set peer** {*hostname* | *ip-address*} **set security-association lifetime seconds** *seconds* and/or **set security-association lifetime kilobytes** *kilobytes* **set pfs** [group1 | group2] exit	Configures a dynamic crypto map using the **crypto dynamic-map** command. For more information on the **crypto dynamic-map** command, refer to the "IPSec Network Security Commands" chapter of the *Cisco IOS Security Command Reference.*
2	**crypto map** *map-name map-number* **ipsec-isakmp dynamic** *dynamic-map-name* [**discover**]	(Optional) Adds a crypto map set to a static crypto map set. For more information on the crypto map command, refer to the "IPSec Network Security Commands" chapter of the *Cisco IOS Security Command Reference.* (Optional) Enter the **discover** keyword on the dynamic crypto map to enable peer discovery. After the dynamic crypto map template permits an outbound packet, peer discovery occurs when the packet reaches an interface configured with the dynamic crypto map.

Clearing IKE Connections

If you want, you can clear existing IKE connections.

To clear IKE connections, use the following commands in EXEC mode:

Step	Command	Purpose
1	**show crypto isakmp sa**	Displays existing IKE connections; note the connection identifiers for connections you wish to clear.
2	clear crypto isakmp [*connection-id*]	Clears IKE connections.

Troubleshooting IKE

To assist in IKE troubleshooting, use the following commands in EXEC mode:

Command	Purpose
show crypto isakmp policy	Displays the parameters for each configured IKE policy.
show crypto isakmp sa	Displays all current IKE security associations.
show running-config	Verifies IKE configuration.
debug crypto isakmp	Displays **debug** messages about IKE events.

What to Do Next

After IKE configuration is complete, you can configure IPSec. IPSec configuration is described in the "Configuring IPSec Network Security" chapter.

IKE Configuration Examples

This example creates two IKE policies, with policy 15 as the highest priority, policy 20 as the next priority, and the existing default priority as the lowest priority. It also creates a pre-shared key to be used with policy 20 with the remote peer whose IP address is 192.168.224.33.

```
crypto isakmp policy 15
 encryption 3des
 hash md5
 authentication rsa-sig
 group 2
 lifetime 5000
crypto isakmp policy 20
 authentication pre-share
 lifetime 10000
crypto isakmp key 1234567890 address 192.168.224.33
```

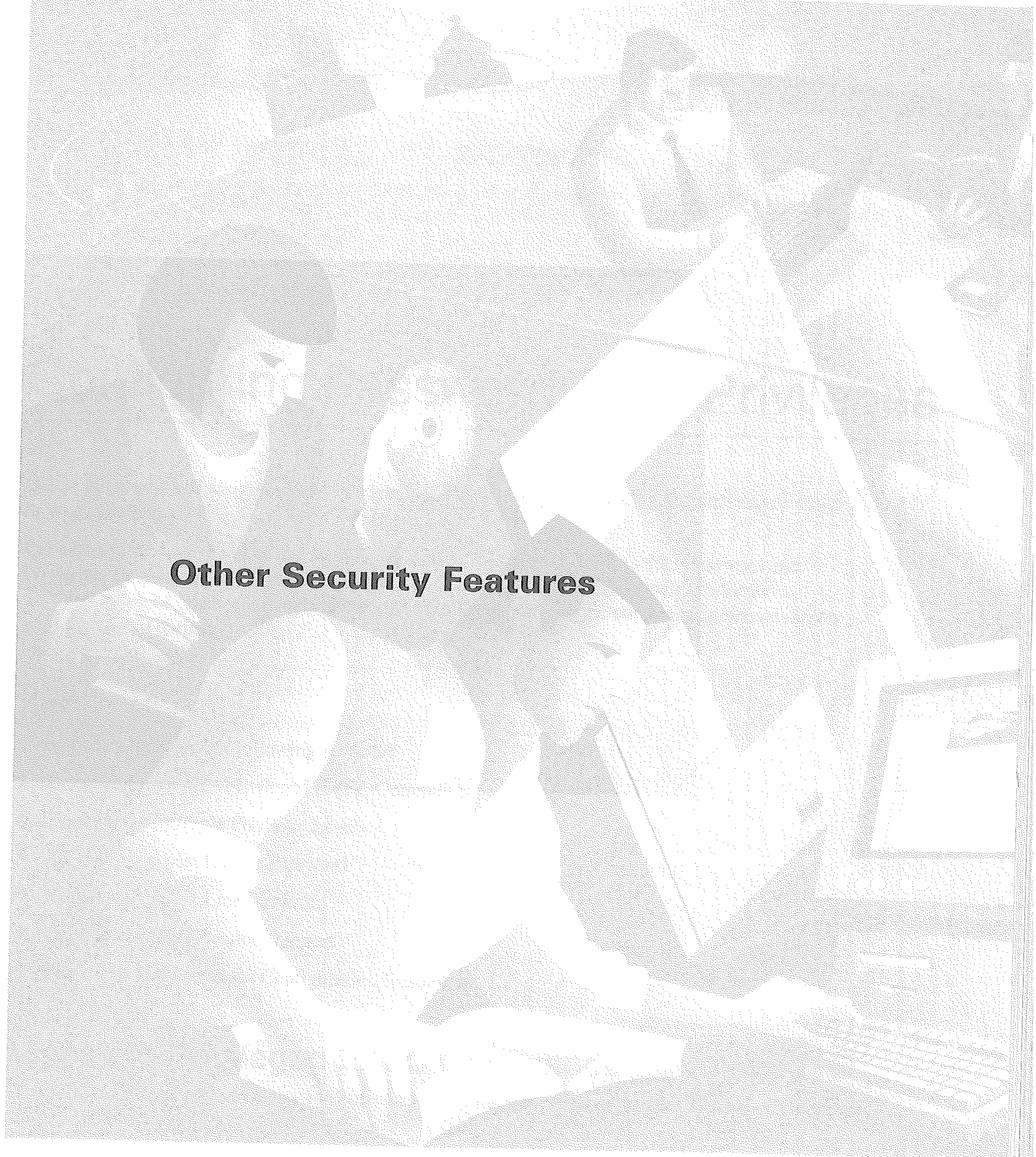
In the preceding example, the **encryption des** of policy 15 would not appear in the written configuration because this is the default value for the encryption algorithm parameter.

If the **show crypto isakmp policy** command is issued with this configuration, the output would be as follows:

```
Protection suite priority 15
encryption algorithm:DES - Data Encryption Standard (56 bit keys)
hash algorithm:Message Digest 5
authentication method:Rivest-Shamir-Adleman Signature
Diffie-Hellman group:#2 (1024 bit)
lifetime:5000 seconds, no volume limit
Protection suite priority 20
encryption algorithm:DES - Data Encryption Standard (56 bit keys)
hash algorithm:Secure Hash Standard
authentication method:Pre-Shared Key
Diffie-Hellman group:#1 (768 bit)
lifetime:10000 seconds, no volume limit
Default protection suite
encryption algorithm:DES - Data Encryption Standard (56 bit keys)
hash algorithm:Secure Hash Standard
authentication method:Rivest-Shamir-Adleman Signature
Diffie-Hellman group:#1 (768 bit)
lifetime:86400 seconds, no volume limit
```

Note that although the output shows "no volume limit" for the lifetimes, you can currently only configure a time lifetime (such as 86,400 seconds); volume limit lifetimes are not configurable.

Other Security Features



Configuring Passwords and Privileges

Using passwords and assigning privilege levels is a simple way of providing terminal access control in your network.

For a complete description of the commands used in this chapter, refer to the "Password and Privileges Commands" chapter in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

In This Chapter

This chapter includes the following sections:

- Protecting Access to Privileged EXEC Commands
- Configuring Multiple Privilege Levels
- Recovering a Lost Enable Password
- Recovering a Lost Line Password
- Configuring Identification Support
- Passwords and Privileges Configuration Examples

Protecting Access to Privileged EXEC Commands

The following tasks provide a way to control access to the system configuration file and privileged EXEC (enable) commands:

- Setting or Changing a Static Enable Password
- Protecting Passwords with Enable Password and Enable Secret
- Setting or Changing a Line Password
- Encrypting Passwords

Setting or Changing a Static Enable Password

To set or change a static password that controls access to privileged EXEC (enable) mode, use the following command in global configuration mode:

Command	Purpose
enable password *password*	Establishes a new password or change an existing password for the privileged command level.

For examples of how to define enable passwords for different privilege levels, see the "Multiple Levels of Privileges Examples" section at the end of this chapter.

Protecting Passwords with Enable Password and Enable Secret

To provide an additional layer of security, particularly for passwords that cross the network or are stored on a TFTP server, you can use either the **enable password** or **enable secret** commands. Both commands accomplish the same thing; that is, they allow you to establish an encrypted password that users must enter to access enable mode (the default), or any privilege level you specify.

We recommend that you use the **enable secret** command because it uses an improved encryption algorithm. Use the **enable password** command only if you boot an older image of the Cisco IOS software, or if you boot older boot ROMs that do not recognize the **enable secret** command.

If you configure the **enable secret** command, it takes precedence over the **enable password** command; the two commands cannot be in effect simultaneously.

To configure the router to require an enable password, use either of the following commands in global configuration mode:

Command	Purpose
enable password [**level** *level*] {*password* | *encryption-type encrypted-password*} or	Establishes a password for a privilege command mode.
enable secret [**level** *level*] {*password* | *encryption-type encrypted-password*}	Specifies a secret password, saved using a non-reversible encryption method. (If enable password and enable secret are both set, users must enter the enable secret password.)

Use either of these commands with the **level** option to define a password for a specific privilege level. After you specify the level and set a password, give the password only to users who need to have access at this level. Use the **privilege level** configuration command to specify commands accessible at various levels.

If you have the **service password-encryption** command enabled, the password you enter is encrypted. When you display it with the **more system:running-config** command, it is displayed in encrypted form.

If you specify an encryption type, you must provide an encrypted password—an encrypted password you copy from another router configuration.

Note You cannot recover a lost encrypted password. You must clear NVRAM and set a new password. See the sections "Recovering a Lost Enable Password" or "Recovering a Lost Line Password" in this chapter if you have lost or forgotten your password.

Setting or Changing a Line Password

To set or change a password on a line, use the following command in global configuration mode:

Command	Purpose
`password password`	Establishes a new password or change an existing password for the privileged command level.

Encrypting Passwords

Because protocol analyzers can examine packets (and read passwords), you can increase access security by configuring the Cisco IOS software to encrypt passwords. Encryption prevents the password from being readable in the configuration file.

To configure the Cisco IOS software to encrypt passwords, use the following command in global configuration mode:

Command	Purpose
`service password-encryption`	Encrypts a password.

The actual encryption process occurs when the current configuration is written or when a password is configured. Password encryption is applied to all passwords, including authentication key passwords, the privileged command password, console and virtual terminal line access passwords, and BGP neighbor passwords. The **service password-encryption** command is primarily useful for keeping unauthorized individuals from viewing your password in your configuration file.

Caution The **service password-encryption** command does not provide a high level of network security. If you use this command, you should also take additional network security measures.

Although you cannot recover a lost encrypted password (that is, you cannot get the original password back), you can recover from a lost encrypted password. See the sections "Recovering a Lost Enable Password" or "Recovering a Lost Line Password" in this chapter if you have lost or forgotten your password.

Configuring Multiple Privilege Levels

By default, the Cisco IOS software has two modes of password security: user mode (EXEC) and privilege mode (enable). You can configure up to 16 hierarchical levels of commands for each mode. By configuring multiple passwords, you can allow different sets of users to have access to specified commands.

For example, if you want the **configure** command to be available to a more restricted set of users than the **clear line** command, you can assign level 2 security to the **clear line** command and distribute the level 2 password fairly widely, and assign level 3 security to the **configure** command and distribute the password to level 3 commands to fewer users.

The following tasks describe how to configure additional levels of security:

- Setting the Privilege Level for a Command
- Changing the Default Privilege Level for Lines
- Displaying Current Privilege Levels
- Logging In to a Privilege Level

Setting the Privilege Level for a Command

To set the privilege level for a command, use the following commands in global configuration mode:

Step	Command	Purpose
1	**privilege mode level** *level command*	Sets the privilege level for a command.
2	**enable password level** *level* [*encryption-type*] *password*	Specifies the enable password for a privilege level.

Changing the Default Privilege Level for Lines

To change the default privilege level for a given line or a group of lines, use the following command in line configuration mode:

Command	Purpose
privilege level *level*	Specifies a default privilege level for a line.

Displaying Current Privilege Levels

To display the current privilege level you can access based on the password you used, use the following command in EXEC mode:

Command	Purpose
show privilege	Displays your current privilege level.

Logging In to a Privilege Level

To log in to a router at a specified privilege level, use the following command in EXEC mode:

Command	Purpose
enable *level*	Logs in to a specified privilege level.

To exit to a specified privilege level, use the following command in EXEC mode:

Command	Purpose
disable *level*	Exits to a specified privilege level.

Recovering a Lost Enable Password

You can restore access to enable mode on a router when the password is lost using one of the three procedures described in this section. The procedure you use depends on your router platform.

You can perform password recovery on most of the platforms without changing hardware jumpers, but all platforms require the configuration to be reloaded. Password recovery can be done only from the console port on the router. Table 27 shows which password recovery procedure to use with each router platform.

Table 27 Platform-Specific Password Recovery Procedures

Password Recovery Procedure	Router Platform
Password Recovery Procedure 1	Cisco 2000 series
	Cisco 2500 series
	Cisco 3000 series
	Cisco 4000 series with 680x0 Motorola CPU
	Cisco 7000 series running Cisco IOS Release 10.0 or later in ROMs installed on the RP card
	IGS series running Cisco Release IOS 9.1 or later in ROMs
Password Recovery Procedure 2	Cisco 1003
	Cisco 1600 series
	Cisco 2600 series
	Cisco 3600 series
	Cisco 4500 series
	Cisco 7100 series
	Cisco 7200 series
	Cisco 7500 series
	IDT Orion-based routers
	AS5200 and AS5300 platforms

This section includes the following sections:

- Password Recovery Process
- Password Recovery Procedure 1
- Password Recovery Procedure 2

Password Recovery Process

Both password recovery procedures involve the following basic steps:

Step 1 Configure the router to boot up without reading the configuration memory (NVRAM). This is sometimes called the test system mode.

Step 2 Reboot the system.

Step 3 Access enable mode (which can be done without a password if you are in test system mode).

Step 4 View or change the password, or erase the configuration.

Step 5 Reconfigure the router to boot up and read the NVRAM as it normally does.

Step 6 Reboot the system.

Note Some password recovery requires that a terminal issue a Break signal; you must be familiar with how your terminal or PC terminal emulator issues this signal. For example, in ProComm, the keys Alt-B by default generates the Break signal, and in a Windows terminal you press Break or CTRL-Break. A Windows terminal also allows you to define a function key as a BREAK signal. To do so, select function keys from the Terminal window and define one as Break by entering the characters **^$B** (**Shift 6**, **Shift 4**, and uppercase **B**).

Password Recovery Procedure 1

Use this procedure to recover lost passwords on the following Cisco routers:

- Cisco 2000 series
- Cisco 2500 series
- Cisco 3000 series
- Cisco 4000 series with 680x0 Motorola CPU
- Cisco 7000 series running Cisco IOS Release 10.0 or later in ROMs installed on the RP card. The router can be booting Cisco IOS Release 10.0 software in Flash memory, but it needs the actual ROMs on the processor card too.
- IGS series running Cisco IOS Release 9.1 or later in ROMs

To recover a password using Procedure 1, perform the following steps:

Step 1 Attach a terminal or PC with terminal emulation software to the console port of the router.

Step 2 Enter the **show version** command and record the setting of the configuration register. It is usually 0x2102 or 0x102.

The configuration register value is on the last line of the display. Note whether the configuration register is set to enable Break or disable Break.

The factory-default configuration register value is 0x2102. Notice that the third digit from the left in this value is 1, which disables Break. If the third digit is *not* 1, Break is enabled.

Step 3 Turn off the router, then turn it on.

Step 4 Press the **Break** key on the terminal within 60 seconds of turning on the router.

The `rommon>` prompt with no router name appears. If it does not appear, the terminal is not sending the correct Break signal. In that case, check the terminal or terminal emulation setup.

Step 5 Enter **o/r0x42** at the `rommon>` prompt to boot from Flash memory or **o/r0x41** to boot from the boot ROMs.

Note The first character is the letter o, not the numeral zero. If you have Flash memory and it is intact, 0x42 is the best setting. Use 0x41 only if the Flash memory is erased or not installed. If you use 0x41, you can only view or erase the configuration. You cannot change the password.

Step 6 At the `rommon>` prompt, enter the initialize command to initialize the router.

This causes the router to reboot but ignore its saved configuration and use the image in Flash memory instead. The system configuration display appears.

Note If you normally use the **boot network** command, or if you have multiple images in Flash memory and you boot a non-default image, the image in Flash might be different.

Step 7 Enter **no** in response to the System Configuration Dialog prompts until the following message appears:

```
Press RETURN to get started!
```

Step 8 Press **Return**.

The `Router>` prompt appears.

Step 9 Enter **enable**.

The `Router#` prompt appears.

Step 10 Choose one of the following options:

- To view the password, if it is not encrypted, enter **more nvram:startup-config**.
- To change the password (if it is encrypted, for example), enter the following commands:

```
Router# configure memory
Router# configure terminal
Router(config)# enable secret 1234abcd
Router(config)# ctrl-z
Router# write memory
```

Note The **enable secret** command provides increased security by storing the enable secret password using a non-reversible cryptographic function; however, you cannot recover a lost password that has been encrypted.

Step 11 Enter **configure terminal** at the EXEC prompt to enter configuration mode.

Step 12 Enter **config-register** and whatever value you recorded in Step 2.

Step 13 Press **Ctrl-Z** to quit from the configuration editor.

Step 14 Enter **reload** at the privileged EXEC prompt and enter **write memory** to save the configuration.

Password Recovery Procedure 2

Use this procedure to recover lost passwords on the following routers:

- Cisco 1003
- Cisco 1600 series
- Cisco 2600 series
- Cisco 3600 series
- Cisco 4500 series
- Cisco 7100 series
- Cisco 7200 series

- Cisco 7500 series
- IDT Orion-Based Routers
- AS5200 and AS5300 platforms

To recover a password using Procedure 2, perform the following steps:

Step 1 Attach a terminal or PC with terminal emulation software to the console port of the router.

Step 2 Enter **show version** and record the setting of the configuration register. It is usually 0x2102 or 0x102.

The configuration register value is on the last line of the display. Note whether the configuration register is set to enable Break or disable Break.

The factory-default configuration register value is 0x2102. Notice that the third digit from the left in this value is 1, which disables Break. If the third digit is *not* 1, Break is enabled.

Step 3 Turn off the router, then turn it on.

Step 4 Press the **Break** key on the terminal within 60 seconds of turning on the router.

The `rommon>` prompt appears. If it does not appear, the terminal is not sending the correct Break signal. In that case, check the terminal or terminal emulation setup.

Step 5 Enter **confreg** at the `rommon>` prompt.

The following prompt appears:

```
Do you wish to change configuration [y/n]?
```

Step 6 Enter **yes** and press **Return**.

Step 7 Enter **no** to subsequent questions until the following prompt appears:

```
ignore system config info [y/n]?
```

Step 8 Enter **yes**.

Step 9 Enter **no** to subsequent questions until the following prompt appears:

```
change boot characteristics [y/n]?
```

Step 10 Enter **yes**.

The following prompt appears:

```
enter to boot:
```

Step 11 At this prompt, either enter **2** and press **Return** if Flash memory or, if Flash memory is erased, enter **1**. If Flash memory is erased, the Cisco 4500 must be returned to Cisco for service. If you enter **1**, you can only view or erase the configuration. You cannot change the password.

A configuration summary is displayed and the following prompt appears:

```
Do you wish to change configuration [y/n]?
```

Step 12 Enter **no** and press **Return**.

The following prompt appears:

```
rommon>
```

Step 13 Enter **reset** at the privileged EXEC prompt or, for Cisco 4500 series and Cisco 7500 series routers, power cycle the router.

Step 14 As the router boots, enter **no** to all the setup questions until the following prompt appears:

```
Router>
```

Step 15 Enter **enable** to enter enable mode.

The `Router#` prompt appears.

Step 16 Choose one of the following options:

- To view the password, if it is not encrypted, enter **more nvram:startup-config**.
- To change the password (if it is encrypted, for example), enter the following commands:

```
Router# configure memory
Router# configure terminal
Router(config)# enable secret 1234abcd
Router(config)# ctrl-z
Router# write memory
```

Note The **enable secret** command provides increased security by storing the enable secret password using a non-reversible cryptographic function; however, you cannot recover a lost password that has been encrypted.

Step 17 Enter **configure terminal** at the prompt.

Step 18 Enter **config-register** and whatever value you recorded in Step 2.

Step 19 Press **Ctrl-Z** to quit from the configuration editor.

Step 20 Enter **reload** at the prompt and enter **write memory** to save the configuration.

Recovering a Lost Line Password

If your router has the nonvolatile memory option, you can accidentally lock yourself out of enable mode if you enable password checking on the console terminal line and then forget the line password. To recover a lost line password, perform the following steps:

Step 1 Force the router into factory diagnostic mode.

See the hardware installation and maintenance publication for your product for specific information about setting the processor configuration register to factory diagnostic mode. Table 28 summarizes the hardware or software settings required by various products to set factory diagnostic mode.

Step 2 Enter **Yes** when asked if you want to set the manufacturers' addresses.

The following prompt appears:

```
TEST-SYSTEM >
```

Step 3 Enter **enable** to enter enable mode:

```
TEST-SYSTEM > enable
```

Step 4 Enter **more nvram:startup-config** to review the system configuration and find the password. Do not change anything in the factory diagnostic mode.

```
TEST-SYSTEM # more nvram:startup-config
```

Step 5 To resume normal operation, restart the router or reset the configuration register.

Step 6 Log in to the router with the password that was shown in the configuration file.

Note All debugging capabilities are turned on during diagnostic mode.

See the hardware installation and maintenance publication for your product for specific information about configuring the processor configuration register for factory diagnostic mode. Table 28 summarizes the hardware or software settings required by the various products to set factory diagnostic mode.

Table 28 Factory Diagnostic Mode Settings for the Configuration Register

Platform	Setting
Modular products	Set jumper in bit 15 of the processor configuration register, then restart; remove the jumper when finished.
Cisco AS5100 Cisco AS5200 Cisco AS5300 Cisco 1600 series Cisco 2500 series Cisco 3000 series Cisco 3600 series Cisco 4000 series Cisco 4500 series Cisco 7000 series Cisco 7100 series Cisco 7200 series Cisco 7500 series	Use the **config-register** command to set the processor configuration register to 0x8000, then initialize and boot the system. Use the **reload** command to restart and set the processor configuration register to 0x2102 when finished.

Configuring Identification Support

Identification support allows you to query a Transmission Control Protocol (TCP) port for identification. This feature enables an unsecure protocol, described in RFC 1413, to report the identity of a client initiating a TCP connection and a host responding to the connection. With identification support, you can connect a TCP port on a host, issue a simple text string to request information, and receive a simple text-string reply.

To configure identification support, use the following command in global configuration mode:

Command	Purpose
`ip identd`	Enables identification support.

Passwords and Privileges Configuration Examples

The following sections provide password and privileges configuration examples:

- Multiple Levels of Privileges Examples
- Username Examples

Multiple Levels of Privileges Examples

This section provides examples of using multiple privilege levels to specify who can access different sets of commands. This section includes the following sections:

- Allowing Users to Clear Lines Examples
- Defining an Enable Password for System Operators Examples
- Disabling a Privilege Level Example

Allowing Users to Clear Lines Examples

If you want to allow users to clear lines, you can do either of the following:

- Change the privilege level for the **clear** and **clear line** commands to 1 or "ordinary user level," as follows. This allows any user to clear lines.

```
privilege exec level 1 clear line
```

- Change the privilege level for the **clear** and **clear line** commands to level 2. To do so, use the **privilege level** global configuration command to specify privilege level 2. Then define an enable password for privilege level 2 and tell only those users who need to know what the password is.

```
enable password level 2 pswd2
privilege exec level 2 clear line
```

Defining an Enable Password for System Operators Examples

In the following example, you define an enable password for privilege level 10 for system operators and make **clear** and **debug** commands available to anyone with that privilege level enabled.

```
enable password level 10 pswd10
privilege exec level 10 clear line
privilege exec level 10 debug ppp chap
privilege exec level 10 debug ppp error
privilege exec level 10 debug ppp negotiation
```

The following example lowers the privilege level of the **more system:running-config** command and most configuration commands to operator level so that the configuration can be viewed by an operator. It leaves the privilege level of the **configure** command at 15. Individual configuration commands are displayed in the **more system:running-config** output only if the privilege level for a command has been lowered to 10. Users are allowed to see only those commands that have a privilege level less than or equal to their current privilege level.

```
enable password level 15 pswd15
privilege exec level 15 configure
enable password level 10 pswd10
privilege exec level 10 more system:running-config
```

Disabling a Privilege Level Example

In the following example, the **show ip route** command is set to privilege level 15. To keep all **show ip** and **show** commands from also being set to privilege level 15, these commands are specified to be privilege level 1.

```
privilege exec level 15 show ip route
privilege exec level 1 show ip
privilege exec level 1 show
```

Username Examples

The following sample configuration sets up secret passwords on Routers A, B, and C, to enable the three routers to connect to each other.

To authenticate connections between Routers A and B, enter the following commands:

On Router A:

```
username B password a-b_secret
```

On Router B:

```
username A password a-b_secret
```

To authenticate connections between Routers A and C, enter the following commands:

On Router A:

```
username C password a-c_secret
```

On Router C:

```
username A password a-c_secret
```

To authenticate connections between Routers B and C, enter the following commands:

On Router B:

```
username C password b-c_secret
```

On Router C:

```
username B password b-c_secret
```

For example, suppose you enter the following command:

```
username bill password westward
```

The system displays this command as follows:

```
username bill password 7 21398211
```

The encrypted version of the password is 21398211. The password was encrypted by the Cisco-defined encryption algorithm, as indicated by the "7."

However, if you enter the following command, the system determines that the password is already encrypted and performs no encryption. Instead, it displays the command exactly as you entered it:

```
username bill password 7 21398211
username bill password 7 21398211
```

Neighbor Router Authentication: Overview and Guidelines

You can prevent your router from receiving fraudulent route updates by configuring neighbor router authentication.

This chapter describes neighbor router authentication as part of a total security plan. It describes what neighbor router authentication is, how it works, and why you should use it to increase your overall network security.

This chapter refers to neighbor router authentication as "neighbor authentication." Neighbor router authentication is also sometimes called "route authentication."

In This Chapter

This chapter describes the following topics:

- Benefits of Neighbor Authentication
- Protocols That Use Neighbor Authentication
- When to Configure Neighbor Authentication
- How Neighbor Authentication Works
- Key Management (Key Chains)
- Finding Neighbor Authentication Configuration Information

Benefits of Neighbor Authentication

When configured, neighbor authentication occurs whenever routing updates are exchanged between neighbor routers. This authentication ensures that a router receives reliable routing information from a trusted source.

Without neighbor authentication, unauthorized or deliberately malicious routing updates could compromise the security of your network traffic. A security compromise could occur if an unfriendly party diverts or analyzes your network traffic. For example, an unauthorized router could send a fictitious routing update to convince your router to send traffic to an incorrect destination. This diverted traffic could be analyzed to learn confidential information of your organization, or merely used to disrupt your organization's ability to effectively communicate using the network.

Neighbor authentication prevents any such fraudulent route updates from being received by your router.

Protocols That Use Neighbor Authentication

Neighbor authentication can be configured for the following routing protocols:

- Border Gateway Protocol (BGP)
- DRP Server Agent
- Intermediate System-to-Intermediate System (IS-IS)
- IP Enhanced Interior Gateway Routing Protocol (IGRP)
- Open Shortest Path First (OSPF)
- Routing Information Protocol (RIP) version 2

When to Configure Neighbor Authentication

You should configure any router for neighbor authentication if that router meets all of these conditions:

- The router uses any of the routing protocols previously mentioned.
- It is conceivable that the router might receive a false route update.
- If the router were to receive a false route update, your network might be compromised.
- If you configure a router for neighbor authentication, you also need to configure the neighbor router for neighbor authentication.

How Neighbor Authentication Works

When neighbor authentication has been configured on a router, the router authenticates the source of each routing update packet that it receives. This is accomplished by the exchange of an authenticating key (sometimes referred to as a password) that is known to both the sending and the receiving router.

There are two types of neighbor authentication used: plain text authentication and Message Digest Algorithm Version 5 (MD5) authentication. Both forms work in the same way, with the exception that MD5 sends a "message digest" instead of the authenticating key itself. The message digest is created using the key and a message, but the key itself is not sent, preventing it from being read while it is being transmitted. Plain text authentication sends the authenticating key itself over the wire.

Note Note that plain text authentication is not recommended for use as part of your security strategy. Its primary use is to avoid accidental changes to the routing infrastructure. Using MD5 authentication, however, is a recommended security practice.

Caution As with all keys, passwords, and other security secrets, it is imperative that you closely guard authenticating keys used in neighbor authentication. The security benefits of this feature are reliant upon your keeping all authenticating keys confidential. Also, when performing router management tasks via Simple Network Management Protocol (SNMP), do not ignore the risk associated with sending keys using non-encrypted SNMP.

This section includes the following sections:

- Plain Text Authentication
- MD5 Authentication

Plain Text Authentication

Each participating neighbor router must share an authenticating key. This key is specified at each router during configuration. Multiple keys can be specified with some protocols; each key must then be identified by a key number.

In general, when a routing update is sent, the following authentication sequence occurs:

Step 1 A router sends a routing update with a key and the corresponding key number to the neighbor router. In protocols that can have only one key, the key number is always zero.

Step 2 The receiving (neighbor) router checks the received key against the same key stored in its own memory.

Step 3 If the two keys match, the receiving router accepts the routing update packet. If the two keys did not match, the routing update packet is rejected.

These protocols use plain text authentication:

- DRP Server Agent
- IS-IS
- OSPF
- RIP version 2

MD5 Authentication

MD5 authentication works similarly to plain text authentication, except that the key is never sent over the wire. Instead, the router uses the MD5 algorithm to produce a "message digest" of the key (also called a "hash"). The message digest is then sent instead of the key itself. This ensures that nobody can eavesdrop on the line and learn keys during transmission.

These protocols use MD5 authentication:

- OSPF
- RIP version 2
- BGP
- IP Enhanced IGRP

Key Management (Key Chains)

You can configure key chains for these IP routing protocols:

- RIP version 2
- IP Enhanced IGRP
- DRP Server Agent

These routing protocols offer the additional function of managing keys by using key chains. When you configure a key chain, you specify a series of keys with lifetimes, and the Cisco IOS software rotates through each of these keys. This decreases the likelihood that keys will be compromised.

Each key definition within the key chain must specify a time interval for which that key will be activated (its "lifetime"). Then, during a given key's lifetime, routing update packets are sent with this activated key. Keys cannot be used during time periods for which they are not activated. Therefore, it is recommended that for a given key chain, key activation times overlap to avoid any period of time for which no key is activated. If a time period occurs during which no key is activated, neighbor authentication cannot occur, and therefore routing updates will fail.

Multiple key chains can be specified.

Note that the router needs to know the time to be able to rotate through keys in synchronization with the other participating routers, so that all routers are using the same key at the same moment. Refer to the Network Time Protocol (NTP) and calendar commands in the "Performing Basic System Management" chapter of the *Cisco IOS Configuration Fundamentals Configuration Guide* for information about configuring time at your router.

Finding Neighbor Authentication Configuration Information

To find complete configuration information for neighbor authentication, refer to the appropriate section and chapter in the *Cisco IOS IP and IP Routing Configuration Guide* as listed in Table 29.

Table 29 Location of Neighbor Authentication Information for Each Supported Protocol

Protocol	Chapter	Section
BGP	"Configuring BGP"	"Configuring Neighbor Options"
DRP Server Agent	"Configuring IP Services"	"Configuring a DRP Server Agent"
IP Enhanced IGRP	"Configuring IP Enhanced IGRP"	"Configuring Enhanced IGRP Route Authentication"
IS-IS	"Configuring Integrated IS-IS"	"Assigning a Password for an Interface" and "Configuring IS-IS Authentication Passwords"
OSPF	"Configuring OSPF"	"Configuring OSPF Interface Parameters" and "Configuring OSPF Area Parameters" and "Creating Virtual Links"
RIP version 2	"Configuring RIP"	"Enabling RIP Authentication"

To find complete configuration information for key chains, refer to the "Managing Authentication Keys" section in the "Configuring IP Routing Protocol-Independent Features" chapter of the *Cisco IOS IP and IP Routing Configuration Guide.*

Configuring IP Security Options

Cisco provides IP Security Option (IPSO) support as described in RFC 1108. Cisco's implementation is only minimally compliant with RFC 1108 because the Cisco IOS software only accepts and generates a 4-byte IPSO.

IPSO is generally used to comply with the U.S. government's Department of Defense security policy.

For a complete description of IPSO commands, refer to the "IP Security Options Commands" chapter of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

In This Chapter

This chapter describes how to configure IPSO for both the basic and extended security options described in RFC 1108. This chapter also describes how to configure auditing for IPSO. This chapter includes the following sections:

- IPSO Configuration Task List
- IPSO Configuration Examples

IPSO Configuration Task List

This section describes the following configuration tasks:

- Configuring Basic IP Security Options
- Configuring Extended IP Security Options
- Configuring the DNSIX Audit Trail Facility

Configuring Basic IP Security Options

Cisco's basic IPSO support provides the following features:

- Defines security level on a per-interface basis
- Defines single-level or multilevel interfaces
- Provides a label for incoming packets
- Strips labels on a per-interface basis
- Reorders options to put any basic security options first

To configure basic IPSO, complete the tasks in the following sections:

- Enabling IPSO and Setting the Security Classifications
- Specifying How IP Security Options Are Processed

Enabling IPSO and Setting the Security Classifications

To enable IPSO and set security classifications on an interface, use either of the following commands in interface configuration mode:

Command	Purpose
ip security dedicated *level authority* [*authority*...]	Sets an interface to the requested IPSO classification and authorities.
ip security multilevel *level1* [*authority1*...] **to** *level2 authority2* [*authority2*...]	Sets an interface to the requested IPSO range of classifications and authorities.

Use the **no ip security** command to reset an interface to its default state.

Specifying How IP Security Options Are Processed

To specify how IP security options are processed, use any of the following optional commands in interface configuration mode:

Command	Purpose
ip security ignore-authorities	Enables an interface to ignore the authorities field of all incoming packets.
ip security implicit-labelling [*level authority* [*authority*...]]	Classifies packets that have no IPSO with an implicit security label.
ip security extended-allowed	Accepts packets on an interface that has an extended security option present.
ip security add	Ensures that all packets leaving the router on an interface contain a basic security option.
ip security strip	Removes any basic security option that might be present on a packet leaving the router through an interface.
ip security first	Prioritizes security options on a packet.
ip security reserved-allowed	Treats as valid any packets that have Reserved1 through Reserved4 security levels.

Default Values for Command Keywords

To fully comply with IPSO, the default values for the minor keywords have become complex. Default value usages include the following:

- The default for all of the minor keywords is *off,* with the exception of **implicit-labelling** and **add**.
- The default value of **implicit-labelling** is *on* if the interface is "unclassified Genser;" otherwise, it is *off.*
- The default value for **add** is *on* if the interface is not "unclassified Genser;" otherwise, it is *off.*

Table 30 provides a list of all default values.

Table 30 Default Security Keyword Values

Interface Type	Level	Authority	Implicit Labeling	Add IPSO
None	None	None	On	Off
Dedicated	Unclassified	Genser	On	Off
Dedicated	Any	Any	Off	On
Multilevel	Any	Any	Off	On

The default value for any interface is "dedicated, unclassified Genser." Note that this implies implicit labeling. This might seem unusual, but it makes the system entirely transparent to packets without options. This is the setting generated when you specify the **no ip security** interface configuration command.

Configuring Extended IP Security Options

Cisco's extended IPSO support is compliant with the Department of Defense Intelligence Information System Network Security for Information Exchange (DNSIX) specification documents. Extended IPSO functionality can unconditionally accept or reject Internet traffic that contains extended security options by comparing those options to configured allowable values. This support allows DNSIX networks to use additional security information to achieve a higher level of security than that achievable with basic IPSO.

Cisco also supports a subset of the security features defined in the DNSIX version 2.1 specification. Specifically, Cisco supports DNSIX definitions of the following:

- How extended IPSO is processed
- Audit trail facility

There are two kinds of extended IPSO fields defined by the DNSIX 2.1 specification and supported by Cisco's implementation of extended IPSO—Network-level Extended Security Option (NLESO) and Auxiliary Extended Security Option (AESO) fields.

NLESO processing requires that security options be checked against configured allowable information, source, and compartment bit values, and requires that the router be capable of inserting extended security options in the IP header.

AESO is similar to NLESO, except that its contents are not checked and are assumed to be valid if its source is listed in the AESO table.

To configure extended IPSO, complete the tasks in the following sections:

- Configuring Global Default Settings
- Attaching ESOs to an Interface
- Attaching AESOs to an Interface

Configuring Global Default Settings

To configure global default setting for extended IPSO, including AESOs, use the following command in global configuration mode:

Command	Purpose
ip security eso-info *source compartment-size default-bit*	Configures system-wide default settings.

Attaching ESOs to an Interface

To specify the minimum and maximum sensitivity levels for an interface, use the following commands in interface configuration mode:

Step	Command	Purpose
1	**ip security eso-min** *source compartment-bits*	Sets the minimum sensitivity level for an interface.
2	**ip security eso-max** *source compartment-bits*	Sets the maximum sensitivity level for an interface.

Attaching AESOs to an Interface

To specify the extended IPSO sources that are to be treated as AESO sources, use the following command in interface configuration mode:

Command	Purpose
ip security aeso *source compartment-bits*	Specifies AESO sources.

DNSIX version 2.1 causes slow-switching code.

See the "IPSO Configuration Examples" section at the end of this chapter.

Configuring the DNSIX Audit Trail Facility

The audit trail facility is a UDP-based protocol that generates an audit trail of IPSO security violations. This facility allows the system to report security failures on incoming and outgoing packets. The Audit Trail Facility sends DNSIX audit trail messages when a datagram is rejected because of IPSO security violations. This feature allows you to configure organization-specific security information.

The DNSIX audit trail facility consists of two protocols:

- DNSIX Message Deliver Protocol (DMDP) provides a basic message-delivery mechanism for all DNSIX elements.
- Network Audit Trail Protocol provides a buffered logging facility for applications to use to generate auditing information. This information is then passed on to DMDP.

To configure the DNSIX auditing facility, complete the tasks in the following sections:

- Enabling the DNSIX Audit Trail Facility
- Specifying Hosts to Receive Audit Trail Messages
- Specifying Transmission Parameters

Enabling the DNSIX Audit Trail Facility

To enable the DNSIX audit trail facility, use the following command in global configuration mode:

Command	Purpose
dnsix-nat source *ip-address*	Starts the audit writing module.

Specifying Hosts to Receive Audit Trail Messages

To define and change primary and secondary addresses of the host to receive audit messages, use the following commands in global configuration mode:

Step	Command	Purpose
1	dnsix-nat primary *ip-address*	Specifies the primary address for the audit trail.
2	dnsix-nat secondary *ip-address*	Specifies the secondary address for the audit trail.
3	**dnsix-nat authorized-redirection** *ip-address*	Specifies the address of a collection center that is authorized to change primary and secondary addresses. Specified hosts are authorized to change the destination of audit messages.

Specifying Transmission Parameters

To specify transmission parameters, use the following commands in global configuration mode:

Step	Command	Purpose
1	**dnsix-nat transmit-count** *count*	Specifies the number of records in a packet before it is sent to a collection center.
2	**dnsix-dmdp retries** *count*	Specifies the number of transmit retries for DMDP.

IPSO Configuration Examples

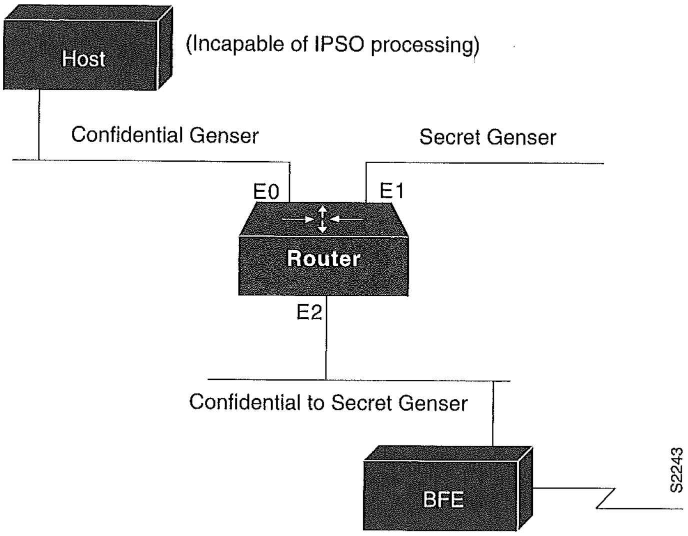
The following sections provide IPSO configuration examples:

- Example 1
- Example 2
- Example 3

Example 1

In this example, three Ethernet interfaces are presented. These interfaces are running at security levels of Confidential Genser, Secret Genser, and Confidential to Secret Genser, as shown in Figure 40.

Figure 40 IPSO Security Levels



The following commands set up interfaces for the configuration in Figure 40:

```
interface ethernet 0
 ip security dedicated confidential genser
interface ethernet 1
 ip security dedicated secret genser
interface ethernet 2
 ip security multilevel confidential genser to secret genser
```

It is possible for the setup to be much more complex.

Example 2

In the following example, there are devices on Ethernet 0 that cannot generate a security option, and so must accept packets without a security option. These hosts do not understand security options; therefore, never place one on such interfaces. Furthermore, there are hosts on the other two networks that are using the extended security option to communicate information, so you must allow these to pass through the system. Finally, there also is a host (a Blacker Front End; see the "Configuring X.25 and LABP" chapter of the *Cisco IOS Wide-Area Networking Configuration Guide* for more information about Blacker emergency mode) on Ethernet 2 that requires the security option to be the first option present, and this condition also must be specified. The new configuration follows.

```
interface ethernet 0
 ip security dedicated confidential genser
 ip security implicit-labelling
 ip security strip
interface ethernet 1
 ip security dedicated secret genser
 ip security extended-allowed
!
interface ethernet 2
 ip security multilevel confidential genser to secret genser
 ip security extended-allowed
 ip security first
```

Example 3

This example shows how to configure a Cisco router with HP-UX CMW DNSIX hosts. The following commands should be configured on each LAN interface of the router for two DNSIX hosts to communicate:

```
ip security multilevel unclassified nsa to top secret nsa
ip security extended allowed
```

DNSIX hosts do not need to know the router's IP addresses, and DNSIX hosts do not need to set up M6RHDB entries for the routers.

Configuring Unicast Reverse Path Forwarding

This chapter describes the Unicast Reverse Path Forwarding (Unicast RPF) feature. The Unicast RPF feature helps to mitigate problems that are caused by malformed or forged IP source addresses that are passing through a router.

For a complete description of the Unicast RPF commands in this chapter, refer to the "Unicast Reverse Path Forwarding Commands" chapter of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

In This Chapter

This chapter has the following sections:

- Feature Overview
- Unicast RPF Configuration Task List
- Troubleshooting Tips
- Monitoring and Maintaining Unicast RPF
- Unicast RPF Configuration Examples

Feature Overview

The Unicast RPF feature helps to mitigate problems that are caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP packets that lack a verifiable IP source address. For example, a number of common types of denial-of-service (DoS) attacks, including Smurf and Tribe Flood Network (TFN), can take advantage of forged or rapidly changing source IP addresses to allow attackers to thwart efforts to locate or filter the attacks. For Internet service providers (ISPs) that provide public access, Unicast RPF deflects such attacks by forwarding only packets that have source addresses that are valid and consistent with the IP routing table. This action protects the network of the ISP, its customer, and the rest of the Internet.

This rest of this section covers the following information:

- How Unicast RPF Works
- Implementing Unicast RPF
- Restrictions
- Related Features and Technologies
- Prerequisites to Configuring Unicast RPF

How Unicast RPF Works

When Unicast RPF is enabled on an interface, the router examines all packets received as input on that interface to make sure that the source address and source interface appear in the routing table and match the interface on which the packet was received. This "look backwards" ability is available only when Cisco express forwarding (CEF) is enabled on the router, because the lookup relies on the presence of the Forwarding Information Base (FIB). CEF generates the FIB as part of its operation.

Note Unicast RPF is an input function and is applied only on the input interface of a router at the upstream end of a connection.

Unicast RPF checks to see if any packet received at a router interface arrives on the best return path (return route) to the source of the packet. Unicast RPF does this by doing a reverse lookup in the CEF table. If the packet was received from one of the best reverse path routes, the packet is forwarded as normal. If there is no reverse path route on the same interface from which the packet was received, it might mean that the source address was modified. If Unicast RPF does not find a reverse path for the packet, the packet is dropped.

Note With Unicast RPF, all equal-cost "best" return paths are considered valid. This means that Unicast RPF works in cases where multiple return paths exist, provided that each path is equal to the others in terms of the routing cost (number of hops, weights, and so on) and as long as the route is in the FIB. Unicast RPF also functions where EIGRP variants are being used and unequal candidate paths back to the source IP address exist.

When a packet is received at the interface where Unicast RPF and access control lists (ACLs) have been configured, the following actions occur:

Step 1 Input ACLs configured on the inbound interface are checked.

Step 2 Unicast RPF checks to see if the packet has arrived on the best return path to the source, which it does by doing a reverse lookup in the FIB table.

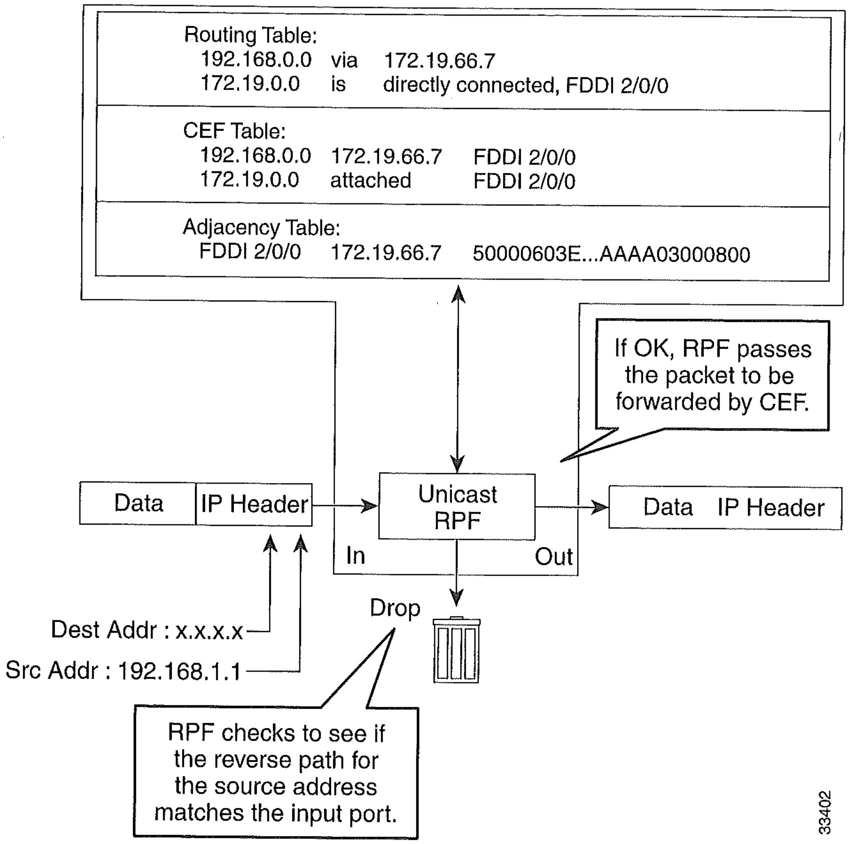
Step 3 CEF table (FIB) lookup is carried out for packet forwarding.

Step 4 Output ACLs are checked on the outbound interface.

Step 5 The packet is forwarded.

Figure 41 illustrates how Unicast RPF and CEF work together to validate IP source addresses by verifying packet return paths. In this example, a customer has sent a packet having a source address of 192.168.1.1 from interface FDDI 2/0/0. Unicast RPF checks the FIB to see if 192.168.1.1 has a path to FDDI 2/0/0. If there is a matching path, the packet is forwarded. If there is no matching path, the packet is dropped.

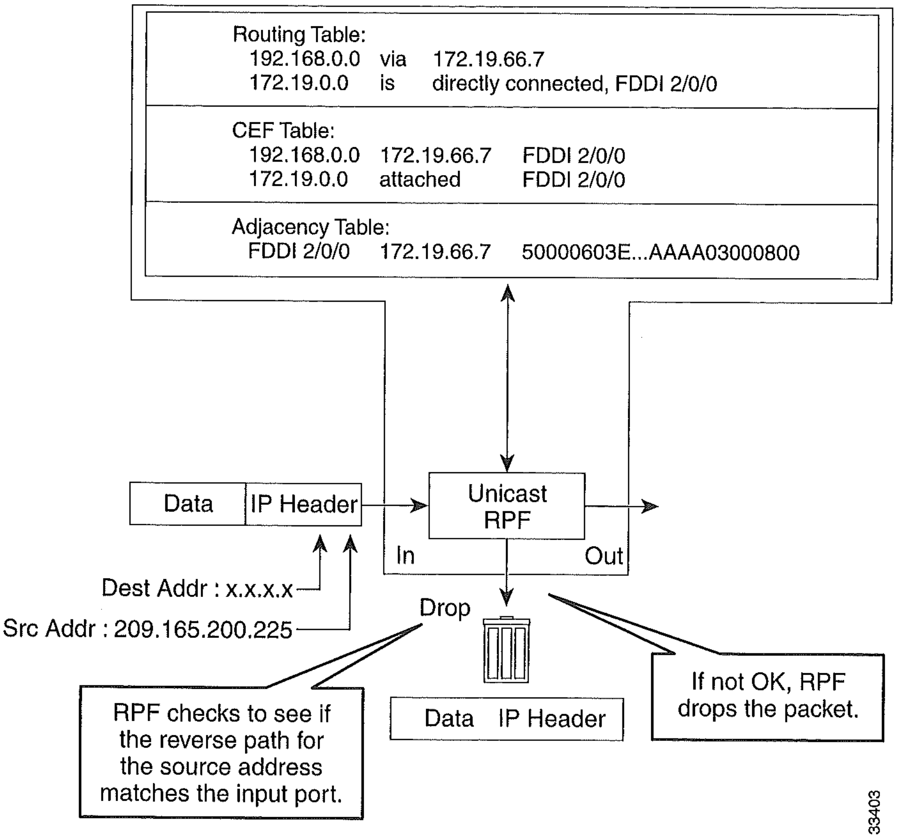
Figure 41 Unicast RPF Validating IP Source Addresses



Routing Table:
192.168.0.0 via 172.19.66.7
172.19.0.0 is directly connected, FDDI 2/0/0
CEF Table:
192.168.0.0 172.19.66.7 FDDI 2/0/0
172.19.0.0 attached FDDI 2/0/0
Adjacency Table:
FDDI 2/0/0 172.19.66.7 50000603E...AAAA03000800
If OK, RPF passes the packet to be forwarded by CEF.
Data
IP Header
Unicast RPF
In
Out
Data IP Header
Drop
Dest Addr : x.x.x.x
Src Addr : 192.168.1.1
RPF checks to see if the reverse path for the source address matches the input port.
33402

Figure 42 illustrates how Unicast RPF drops packets that fail validation. In this example, a customer has sent a packet having a source address of 209.165.200.225, which is received at interface FDDI 2/0/0. Unicast RPF checks the FIB to see if 209.165.200.225 has a return path to FDDI 2/0/0. If there is a matching path, the packet is forwarded. In this case, there is no reverse entry in the routing table that routes the customer packet back to source address 209.165.200.225 on interface FDDI 2/0/0, and so the packet is dropped.

Figure 42 Unicast RPF Dropping Packets That Fail Verification


Routing Table:
192.168.0.0 via 172.19.66.7
172.19.0.0 is directly connected, FDDI 2/0/0
CEF Table:
192.168.0.0 172.19.66.7 FDDI 2/0/0
172.19.0.0 attached FDDI 2/0/0
Adjacency Table:
FDDI 2/0/0 172.19.66.7 50000603E...AAAA03000800
Data
IP Header
Unicast RPF
In
Out
Dest Addr : x.x.x.x
Src Addr : 209.165.200.225
Drop
Data IP Header
RPF checks to see if the reverse path for the source address matches the input port.
If not OK, RPF drops the packet.
33403

Implementing Unicast RPF

Unicast RPF has several key implementation principles:

- The packet must be received at an interface that has the best return path (route) to the packet source (a process called *symmetric routing*). There must be a route in the FIB matching the route to the receiving interface. Adding a route in the FIB can be done via static route, network statement, or dynamic routing.
- IP source addresses at the receiving interface must match the routing entry for the interface.
- Unicast RPF is an input function and is applied only on the input interface of a router at the upstream end of a connection.

Given these implementation principles, Unicast RPF becomes a tool that network administrators can use not only for their customers but also for their downstream network or ISP, even if the downstream network or ISP has other connections to the Internet.

Caution Using optional BGP attributes such as weight, local preference, and so on, the best path back to the source address can be modified, which would affect the operation of Unicast RPF.

This section provides information about the implementation of Unicast RPF:

- Security Policy and Unicast RPF
- Where to Use Unicast RPF
- Routing Table Requirements
- Where Not to Use Unicast RPF
- Unicast RPF with BOOTP and DHCP

Security Policy and Unicast RPF

Consider the following points in determining your policy for deploying Unicast RPF:

- Unicast RPF must be applied at the interface downstream from the larger portion of the network, preferably at the edges of your network.
- The further downstream you apply Unicast RPF, the finer the granularity you have in mitigating address spoofing and in identifying the sources of spoofed addresses. For example, applying Unicast RPF on an aggregation router helps mitigate attacks from many downstream networks or clients and is simple to administer, but it does not help identify the source of the attack. Applying Unicast RPF at the network access server helps limit the scope of the attack and trace the source of the attack; however, deploying Unicast RPF across many sites does add to the administration cost of operating the network.
- The more entities that deploy Unicast RPF across Internet, intranet, and extranet resources, the better the chances of mitigating large-scale network disruptions throughout the Internet community, and the better the chances of tracing the source of an attack.
- Unicast RPF will not inspect IP packets encapsulated in tunnels, such as GRE, LT2P, or PPTP. Unicast RPF must be configured at a home gateway so that Unicast RPF processes network traffic only after the tunneling and encryption layers have been stripped off the packets.

Where to Use Unicast RPF

Unicast RPF can be used in any "single-homed" environment where there is essentially only one access point out of the network; that is, one upstream connection. Networks having one access point offer the best example of symmetric routing, which means that the interface where a packet enters the network is also the best return path to the source of the IP packet. Unicast RPF is best used at the network perimeter for Internet, intranet, or extranet environments, or in ISP environments for customer network terminations.

The following sections provide a look at implementing Unicast RPF in two network environments:

- Enterprise Networks with a Single Connection to an ISP
- Network Access Server Application: Applying Unicast RPF in PSTN/ISDN PoP Aggregation Routers

Enterprise Networks with a Single Connection to an ISP

In enterprise networks, one objective of using Unicast RPF for filtering traffic at the input interface (a process called *ingress filtering*) is for protection from malformed packets arriving from the Internet. Traditionally, local networks that have one connection to the Internet would use ACLs at the receiving interface to prevent spoofed packets from the Internet from entering their local network.

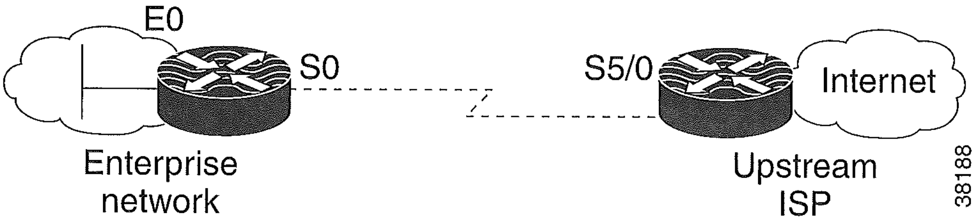
ACLs work well for many single-homed customers; however, there are trade-offs when ACLs are used as ingress filters, including two commonly referenced limitations:

- Packet per second (PPS) performance at very high packet rates
- Maintenance of the ACL (whenever there are new addresses added to the network)

Unicast RPF is one tool that addresses both of these limitations. With Unicast RPF, ingress filtering is done at CEF PPS rates. This processing speed makes a difference when the link is more than 1 Mbps. Additionally, since Unicast RPF uses the FIB, no ACL maintenance is necessary, and thus the administration overhead of traditional ACLs is reduced. The following figure and example demonstrate how this is configured.

Figure 43 illustrates an enterprise network that has a single link to an upstream ISP. In this example, Unicast RPF is applied at interface S0 on the enterprise router for protection from malformed packets arriving from the Internet. Unicast RPF is also applied at interface S5/0 on the ISP router for protection from malformed packets arriving from the enterprise network.

Figure 43 Enterprise Network Using Unicast RPF for Ingress Filtering



Using Figure 43, a typical configuration (assuming that CEF is turned on) on the ISP router would be as follows:

```
ip cef
interface loopback 0
  description Loopback interface on Gateway Router 2
  ip address 192.168.3.1 255.255.255.255
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
interface Serial 5/0
  description 128K HDLC link to ExampleCorp WT50314E  R5-0
  bandwidth 128
  ip unnumbered loopback 0
  ip verify unicast reverse-path
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
ip route 192.168.10.0 255.255.252.0 Serial 5/0
```

The gateway router configuration of the enterprise network (assuming that CEF is turned on) would look similar to the following:

```
ip cef
interface Ethernet 0
  description ExampleCorp LAN
  ip address 192.168.10.1 255.255.252.0
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
interface Serial 0
  description 128K HDLC link to ExampleCorp Internet Inc WT50314E  C0
  bandwidth 128
  ip unnumbered ethernet 0
  ip verify unicast reverse-path
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
ip route 0.0.0.0 0.0.0.0 Serial 0
```

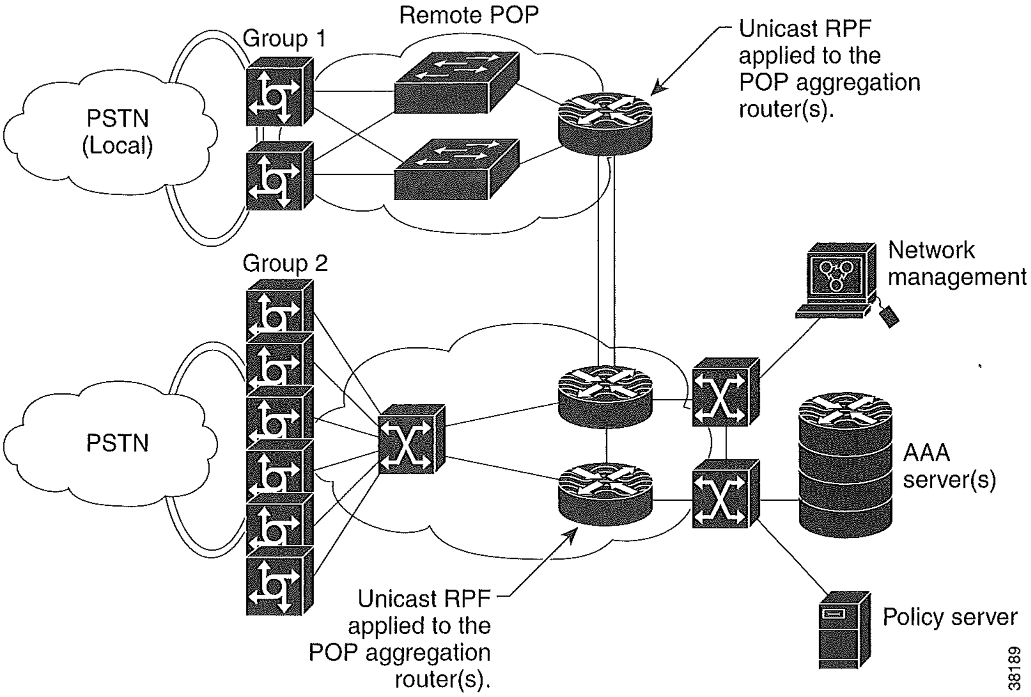
Notice that Unicast RPF works with a single default route. There are no additional routes or routing protocols. Network 192.168.10.0/22 is a connected network. Hence, packets coming from the Internet with a source address in the range 192.168.10.0/22 will be dropped by Unicast RPF.

Network Access Server Application: Applying Unicast RPF in PSTN/ISDN PoP Aggregation Routers

Aggregation routers are ideal places to use Unicast RPF with single-homed clients. Unicast RPF works equally well on leased-line or PSTN/ISDN/xDSL customer connections into the Internet. In fact, dialup connections are reputed to be the greatest source of DoS attacks using forged IP addresses. As long as the network access server supports CEF, Unicast RPF will work. In this topology, the customer aggregation routers need not have the full Internet routing table. Aggregation routers need the routing prefixes information (IP address block); hence, information configured or redistributed in the Interior Gateway Protocol (IGP) or Internal Border Gateway Protocol (IBGP) (depending on the way that you add customer routes into your network) would be enough for Unicast RPF to do its job.

Figure 44 illustrates the application of Unicast RPF to the aggregation and access routers for an Internet service provider (ISP) point-of-presence (POP), with the ISP routers providing dialup customer connections. In this example, Unicast RPF is applied upstream from the customer dialup connection router on the receiving (input) interfaces of the ISP aggregation routers.

Figure 44 Unicast RPF Applied to PSTN/ISDN Customer Connections


Remote POP
Group 1
PSTN (Local)
Unicast RPF applied to the POP aggregation router(s).
Group 2
Network management
PSTN
AAA server(s)
Policy server
Unicast RPF applied to the POP aggregation router(s).
38189

Routing Table Requirements

To work correctly, Unicast RPF needs proper information in the CEF tables. This requirement does not mean that the router must have the entire Internet routing table. The amount of routing information needed in the CEF tables depends on where Unicast RPF is configured and what functions the router performs in the network. For example, in an ISP environment, a router that is a leased-line aggregation router for customers needs only the information based on the static routes redistributed into the IGP or IBGP (depending on which technique is used in the network). Unicast RPF would be configured on the customer interfaces—hence the requirement for minimal routing information. In another scenario, a single-homed ISP could place Unicast RPF on the gateway link to the Internet. The full Internet routing table would be required. Requiring the full routing table would help protect the ISP from external DoS attacks that use addresses that are not in the Internet routing table.

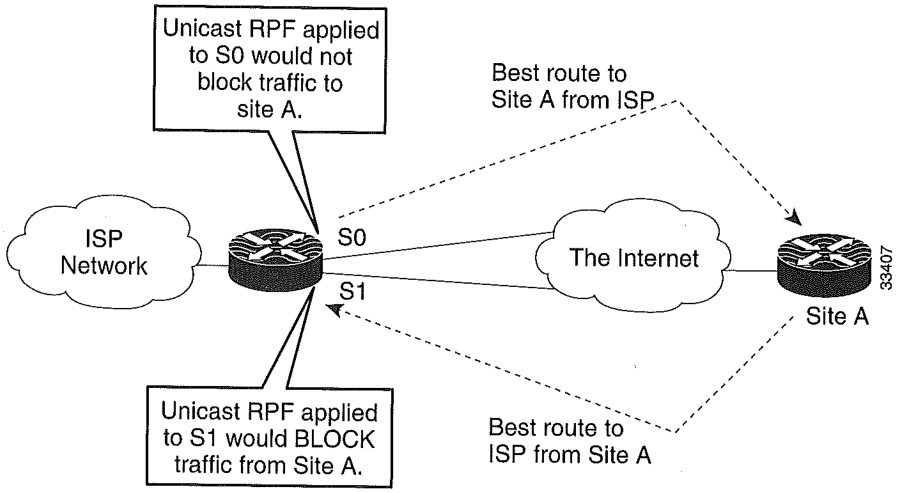
Where Not to Use Unicast RPF

Unicast RPF should not be used on interfaces that are internal to the network. Internal interfaces are likely to have routing asymmetry (see Figure 45), meaning multiple routes to the source of a packet. Unicast RPF should be applied only where there is natural or configured symmetry. As long as administrators carefully plan which interfaces they activate Unicast RPF on, routing asymmetry is not a serious problem.

For example, routers at the edge of the network of an ISP are more likely to have symmetrical reverse paths than routers that are in the core of the ISP network. Routers that are in the core of the ISP network have no guarantee that the best forwarding path out of the router will be the path selected for packets returning to the router. Hence, it is not recommended that you apply Unicast RPF where there is a chance of asymmetric routing. It is simplest to place Unicast RPF only at the edge of a network or, for an ISP, at the customer edge of the network.

Figure 45 illustrates how Unicast RPF can block legitimate traffic in an asymmetrical routing environment.

Figure 45 Unicast RPF Blocking Traffic in an Asymmetrical Routing Environment


Unicast RPF applied to S0 would not block traffic to site A.
Best route to Site A from ISP
ISP Network
S0
S1
The Internet
Site A
33407
Unicast RPF applied to S1 would BLOCK traffic from Site A.
Best route to ISP from Site A

Unicast RPF with BOOTP and DHCP

Unicast RPF will allow packets with 0.0.0.0 source and 255.255.255.255 destination to pass so that Bootstrap Protocol (BOOTP) and Dynamic Host Configuration Protocol (DHCP) functions work properly. This enhancement was added in Cisco IOS Release 12.0 and later, but is not in Cisco IOS Release 11.1CC.

Restrictions

There are some basic restrictions to applying Unicast RPF to multihomed clients:

- Clients should not be multihomed to the same router because multihoming defeats the purpose of building a redundant service for the client.
- Customers must ensure that the packets flowing up the link (out to the Internet) match the route advertised out the link. Otherwise, Unicast RPF filters those packets as malformed packets.
- Unicast RPF is available only for platform images that support CEF. Unicast RPF is supported in Cisco IOS Releases 11.1(17)CC and 12.0 and later. It is not available in Cisco IOS Releases 11.2 or 11.3.

Related Features and Technologies

For more information about Unicast RPF related features and technologies, review the following:

- Unicast RPF requires Cisco express forwarding (CEF) to function properly on the router. For more information about CEF, refer to the *Cisco IOS Switching Services Configuration Guide.*
- Unicast RPF can be more effective at mitigating spoofing attacks when combined with a policy of *ingress* and *egress* filtering using Cisco IOS access control lists (ACLs).
  - — Ingress filtering applies filters to traffic received at a network interface from either internal or external networks. With ingress filtering, packets that arrive from other networks or the Internet and that have a source address that matches a local network, private, or broadcast address are dropped. In ISP environments, for example, ingress filtering can apply to traffic received at the router from either the client (customer) or the Internet.
  - — Egress filtering applies filters to traffic exiting a network interface (the sending interface). By filtering packets on routers that connect your network to the Internet or to other networks, you can permit only packets with valid source IP addresses to leave your network.

  For more information on network filtering, refer to RFC 2267 and to the *Cisco IOS IP and IP Routing Configuration Guide.*
- Cisco IOS software provides additional features that can help mitigate DoS attacks:
  - — Committed Access Rate (CAR). CAR allows you to enforce a bandwidth policy against network traffic that matches an access list. For example, CAR allows you to rate-limit what should be low-volume traffic, such as ICMP traffic. To find out more about CAR, refer to the *Cisco IOS Quality of Service Solutions Configuration Guide.*
  - — Context-based Access Control (CBAC). CBAC selectively blocks any network traffic not originated by a protected network. CBAC uses timeout and threshold values to manage session state information, helping to determine when to drop sessions that do not become fully established. Setting timeout values for network sessions helps mitigate DoS attacks by freeing up system resources, dropping sessions after a specified amount of time. For more information on CBAC, refer to the *Cisco IOS Security Configuration Guide.*
  - — TCP Intercept. The TCP Intercept feature implements software to protect TCP servers from TCP SYN-flooding attacks, which are a type of DoS attack. A SYN-flooding attack occurs when a hacker floods a server with a barrage of requests for connection. Like CBAC, the TCP Intercept feature also uses timeout and threshold values to manage session state information, helping to determine when to drop sessions that do not become fully established. For more information on TCP Intercept, refer to the *Cisco IOS Security Configuration Guide.*

Prerequisites to Configuring Unicast RPF

Prior to configuring Unicast RPF, configure ACLs:

- Configure ACLs to mitigate transmission of invalid IP addresses (perform egress filtering). Permit only valid source addresses to leave your network and get onto the Internet. Prevent all other source addresses from leaving your network for the Internet.
- Configure ACLs to prevent (deny) reception of invalid IP addresses (perform ingress filtering). Invalid addresses include the following types:
  - — Reserved addresses
  - — Loopback addresses
  - — Private addresses

- — Broadcast addresses
- — Source addresses that match any addresses on the protected network (prevents spoofing)

Unicast RPF Configuration Task List

The following sections describe the configuration tasks for Unicast RPF. Each task in the list is identified as either optional or required.

- Configuring Unicast RPF (Required)
- Verifying Unicast RPF (Optional)

See the end of this chapter for the section "Unicast RPF Configuration Examples."

Configuring Unicast RPF

To use Unicast RPF, you must configure the router for CEF switching or CEF distributed switching. There is no need to configure the input interface for CEF switching because Unicast RPF has been implemented as a search through the FIB using the source IP address. As long as CEF is running on the router, individual interfaces can be configured with other switching modes. Unicast RPF is an input-side function that is enabled on an interface or subinterface that supports any type of encapsulation and operates on IP packets received by the router. It is very important that CEF be turned on globally in the router—Unicast RPF will not work without CEF.

To configure Unicast RPF, use the following commands beginning in global configuration mode

Step	Command	Purpose

1	`Router(config)# ip cef`
	`or`
	`Router(config)# ip cef distributed`
		Enables CEF or distributed CEF on the router. Distributed CEF is required for routers that use a Route/Switch Processor (RSP) and Versatile Interface Processor (VIP), which includes Unicast RPF.
		You might want to disable CEF or distributed CEF (dCEF) on a particular interface if that interface is configured with a feature that CEF or dCEF does not support. In this case, you would enable CEF globally, but disable CEF on a specific interface using the **no ip route-cache cef** interface command, which enables all but that specific interface to use express forwarding. If you have disabled CEF or dCEF operation on an interface and want to reenable it, you can do so by using the **ip route-cache cef** command in interface configuration mode.

2	`Router(config-if)# interface` *`type`*	Selects the input interface on which you want to apply Unicast RPF. This is the receiving interface, which allows Unicast RPF to verify the best return path before forwarding the packet on to the next destination. The interface type is specific to your router and the types of interface cards installed on the router. To display a list of available interface types, enter the **interface ?** command.

Step	Command	Purpose
3	`Router(config-if)# ip verify unicast reverse-path`	Enables Unicast RPF on the interface.
4	`Router(config-if)# exit`	Exits interface configuration mode. Repeat Steps 2 and 3 for each interface on which you want to apply Unicast RPF.

Verifying Unicast RPF

To verify that Unicast RPF is operational, use the **show cef interface** command. The following example shows that Unicast RPF is enabled at interface serial2/0/0.

```
Router-3# show cef interface serial 2/0/0
Serial2/0/0 is up (if_number 8)
  Internet address is 192.168.10.2/30
  ICMP redirects are never sent
  Per packet loadbalancing is disabled
  IP unicast RPF check is enabled
  Inbound access list is not set
  Outbound access list is not set
  Interface is marked as point to point interface
  Packets switched to this interface on linecard are dropped to next slow path
  Hardware idb is Serial2/0/0
  Fast switching type 4, interface type 6
  IP Distributed CEF switching enabled
  IP LES Feature Fast switching turbo vector
  IP Feature CEF switching turbo vector
  Input fast flags 0x40, Output fast flags 0x0, ifindex 7(7)
  Slot 2 Slot unit 0 VC -1
  Transmit limit accumulator 0x48001A02 (0x48001A02)
  IP MTU 1500
```

Troubleshooting Tips

If you experience problems while using Unicast RPF, check the following:

HSRP Failure

Failure to disable Unicast RPF before disabling CEF can cause Hot Standby Router Protocol (HSRP) failure. If you want to disable CEF on the router, you must first disable Unicast RPF. To disable Unicast RPF, see the section "Monitoring and Maintaining Unicast RPF."

Dropped Boot Requests

In Cisco IOS Release 11.1(17)CC, Unicast RPF can drop BOOTP request packets that have a source address of 0.0.0.0 due to source address verification at the interface. To enable boot requests to work on the interface, you must use ACLs instead of Unicast RPF.

Monitoring and Maintaining Unicast RPF

This section describes commands used to monitor and maintain Unicast RPF.

Command	Purpose
`Router# show ip traffic`	Displays statistics about IP traffic.
`Router(config-if)# no ip verify unicast reverse-path`	Disables Unicast RPF at the interface.

Caution To disable CEF, you must first disable Unicast RPF. Failure to disable Unicast RPF before disabling CEF can cause HSRP failure. If you want to disable CEF on the router, you must first disable Unicast RPF.

A counter is maintained to count the number of packets discarded because of Unicast RPF. The value of the counter is displayed as part of the output from the **show ip traffic** command. The value of the counter is the total of dropped packets for all router interfaces. The Unicast RPF drop count is included in the IP statistics section.

```
Router# show ip traffic
IP statistics:
  Rcvd:  1471590 total, 887368 local destination
         0 format errors, 0 checksum errors, 301274 bad hop count
         0 unknown protocol, 0 not a gateway
         0 security failures, 0 bad options, 0 with options
  Opts:  0 end, 0 nop, 0 basic security, 0 loose source route
         0 timestamp, 0 extended security, 0 record route
         0 stream ID, 0 strict source route, 0 alert, 0 other
  Frags: 0 reassembled, 0 timeouts, 0 couldn't reassemble
         0 fragmented, 0 couldn't fragment
  Bcast: 205233 received, 0 sent
  Mcast: 463292 received, 462118 sent
  Sent:  990158 generated, 282938 forwarded
  Drop:  3 encapsulation failed, 0 unresolved, 0 no adjacency
         0 no route, 0 unicast RPF, 0 forced drop
```

If the drop counter (router drop count) is not zero, a value indicates that packets were dropped by Unicast RPF. Dropped packets can mean two things:

- Unicast RPF is dropping packets that have a bad source address (normal operation).
- Unicast RPF is dropping legitimate packets because the route is misconfigured to use Unicast RPF in environments where asymmetric routing exists; that is, where multiple paths can exist as the best return path for a source address.

Unicast RPF Configuration Examples

This section provides the following configuration examples:

- Unicast RPF on a Leased-Line Aggregation Router Example
- Unicast RPF on the Cisco AS5800 Using Dialup Ports Example
- Unicast RPF with Inbound and Outbound Filters Example

Unicast RPF on a Leased-Line Aggregation Router Example

The following commands enable Unicast RPF on a serial interface.

```
ip cef
! or "ip cef distributed" for RSP+VIP based routers
!
interface serial 5/0/0
 ip verify unicast reverse-path
```

Unicast RPF on the Cisco AS5800 Using Dialup Ports Example

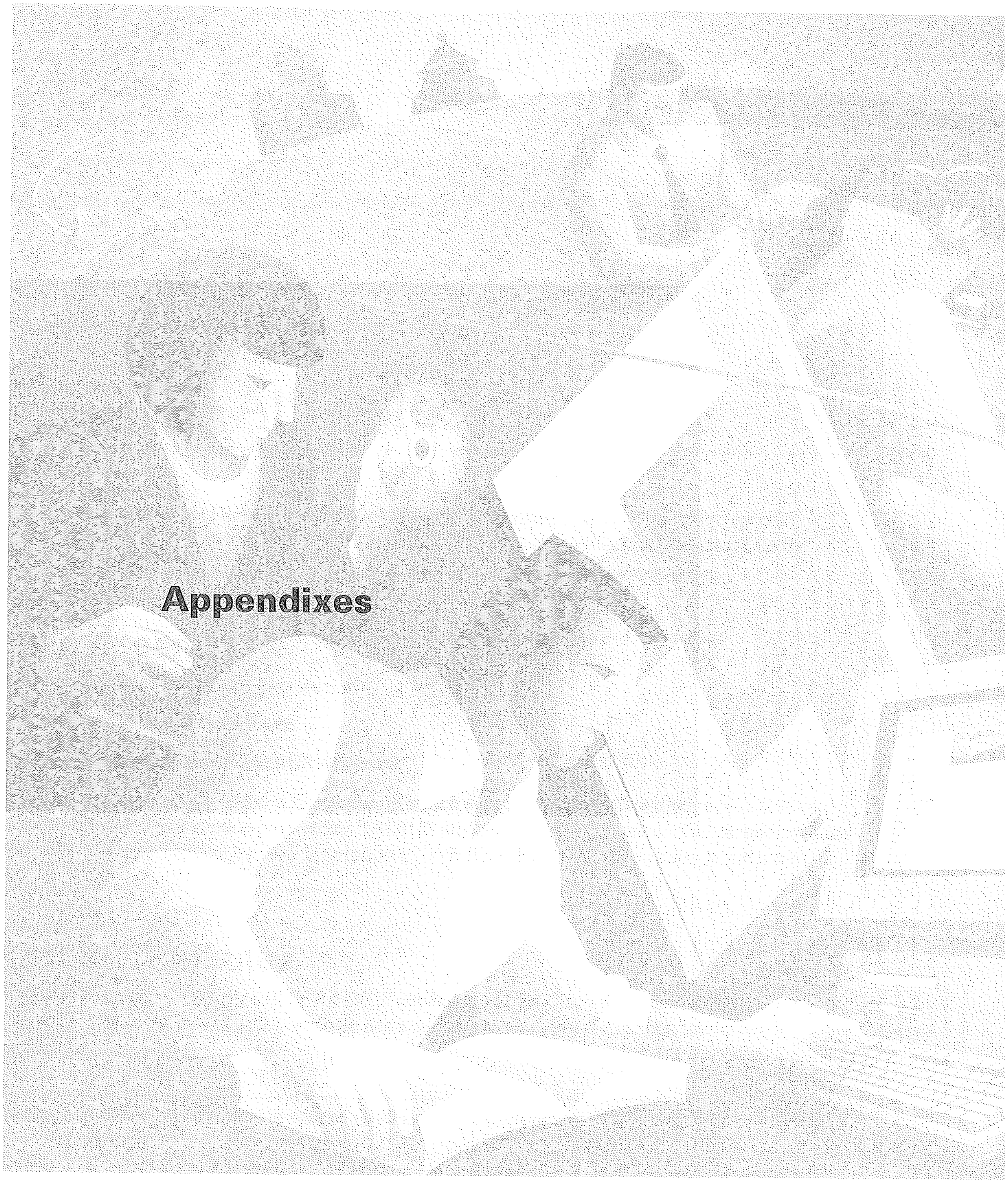
The following example enables Unicast RPF on a Cisco AS5800. The **interface Group-Async** command makes it easy to apply Unicast RPF on all the dialup ports.

```
ip cef
!
interface Group-Async1
 ip verify unicast reverse-path
```

Unicast RPF with Inbound and Outbound Filters Example

The following example uses a very simple single-homed ISP to demonstrate the concepts of ingress and egress filters used in conjunction with Unicast RPF. The example illustrates an ISP-allocated classless interdomain routing (CIDR) block 209.165.202.128/28 that has both inbound and outbound filters on the upstream interface. Be aware that ISPs are usually not single-homed. Hence, provisions for asymmetrical flows (when outbound traffic goes out one link and returns via a different link) need to be designed into the filters on the border routers of the ISP.

```
ip cef distributed
!
interface Serial 5/0/0
 description Connection to Upstream ISP
 ip address 209.165.200.225 255.255.255.252
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip verify unicast reverse-path
 ip access-group 111 in
 ip access-group 110 out
!
access-list 110 permit ip 209.165.202.128 0.0.0.31 any
access-list 110 deny ip any any log
access-list 111 deny ip host 0.0.0.0 any log
access-list 111 deny ip 127.0.0.0 0.255.255.255 any log
access-list 111 deny ip 10.0.0.0 0.255.255.255 any log
access-list 111 deny ip 172.16.0.0 0.15.255.255 any log
access-list 111 deny ip 192.168.0.0 0.0.255.255 any log
access-list 111 deny ip 209.165.202.128 0.0.0.31 any log
access-list 111 permit ip any any
```



Appendixes

RADIUS Attributes

Remote Authentication Dial-In User Service (RADIUS) attributes are used to define specific authentication, authorization, and accounting elements in a user profile, which is stored on the RADIUS daemon. This appendix lists the RADIUS attributes currently supported.

How to Use This Appendix

This appendix is divided into two sections:

- Supported RADIUS Attributes
- Comprehensive List of RADIUS Attributes

The first section lists the Cisco IOS releases in which supported Internet Engineering Task Force (IETF) RADIUS and vendor-proprietary RADIUS attributes are implemented. The second section provides a comprehensive list and description of both IETF RADIUS and vendor-proprietary RADIUS attributes.

Supported RADIUS Attributes

Table 31 lists Cisco-supported IETF RADIUS attributes and the Cisco IOS release in which they are implemented. In cases where the attribute has a security server-specific format, the format is specified.

Note Attributes implemented in special (AA) or early development (T) releases will be added to the next mainline image.

Table 31 Supported RADIUS IETF Attributes

Number	IETF Attribute	11.1	11.2	11.3	11.3 AA	11.3T	12.0	12.1
1	User-Name	yes	yes	yes	yes	yes	yes	yes
2	User-Password	yes	yes	yes	yes	yes	yes	yes
3	CHAP-Password	yes	yes	yes	yes	yes	yes	yes
4	NAS-IP Address	yes	yes	yes	yes	yes	yes	yes
5	NAS-Port	yes	yes	yes	yes	yes	yes	yes
6	Service-Type	yes	yes	yes	yes	yes	yes	yes

Table 31 Supported RADIUS IETF Attributes (continued)

Number	IETF Attribute	11.1	11.2	11.3	11.3 AA	11.3T	12.0	12.1
7	Framed-Protocol	yes	yes	yes	yes	yes	yes	yes
8	Framed-IP-Address	yes	yes	yes	yes	yes	yes	yes
9	Framed-IP-Netmask	yes	yes	yes	yes	yes	yes	yes
10	Framed-Routing	yes	yes	yes	yes	yes	yes	yes
11	Filter-Id	yes	yes	yes	yes	yes	yes	yes
12	Framed-MTU	yes	yes	yes	yes	yes	yes	yes
13	Framed-Compression	yes	yes	yes	yes	yes	yes	yes
14	Login-IP-Host	yes	yes	yes	yes	yes	yes	yes
15	Login-Service	yes	yes	yes	yes	yes	yes	yes
16	Login-TCP-Port	yes	yes	yes	yes	yes	yes	yes
18	Reply-Message	yes	yes	yes	yes	yes	yes	yes
19	Callback-Number	no	no	no	no	no	no	yes
20	Callback-ID	no	no	no	no	no	no	no
22	Framed-Route	yes	yes	yes	yes	yes	yes	yes
23	Framed-IPX-Network	no	no	no	no	no	no	no
24	State	yes	yes	yes	yes	yes	yes	yes
25	Class	yes	yes	yes	yes	yes	yes	yes
26	Vendor-Specific	yes	yes	yes	yes	yes	yes	yes
27	Session-Timeout	yes	yes	yes	yes	yes	yes	yes
28	Idle-Timeout	yes	yes	yes	yes	yes	yes	yes
29	Termination-Action	no	no	no	no	no	no	yes
30	Called-Station-Id	yes	yes	yes	yes	yes	yes	yes
31	Calling-Station-Id	yes	yes	yes	yes	yes	yes	yes
32	NAS-Identifier	no	no	no	no	no	no	no
33	Proxy-State	no	no	no	no	no	no	no
34	Login-LAT-Service	yes	yes	yes	yes	yes	yes	yes
35	Login-LAT-Node	no	no	no	no	no	no	no
36	Login-LAT-Group	no	no	no	no	no	no	no
37	Framed-AppleTalk-Link	no	no	no	no	no	no	no
38	Framed-AppleTalk-Network	no	no	no	no	no	no	no
39	Framed-AppleTalk-Zone	no	no	no	no	no	no	no
40	Acct-Status-Type	yes	yes	yes	yes	yes	yes	yes
41	Acct-Delay-Time	yes	yes	yes	yes	yes	yes	yes
42	Acct-Input-Octets	yes	yes	yes	yes	yes	yes	yes
43	Acct-Output-Octets	yes	yes	yes	yes	yes	yes	yes
44	Acct-Session-Id	yes	yes	yes	yes	yes	yes	yes
45	Acct-Authentic	yes	yes	yes	yes	yes	yes	yes
46	Acct-Session-Time	yes	yes	yes	yes	yes	yes	yes

Table 31 Supported RADIUS IETF Attributes (continued)

Number	IETF Attribute	11.1	11.2	11.3	11.3 AA	11.3T	12.0	12.1
47	Acct-Input-Packets	yes	yes	yes	yes	yes	yes	yes
48	Acct-Output-Packets	yes	yes	yes	yes	yes	yes	yes
49	Acct-Terminate-Cause	no	no	no	yes	yes	yes	yes
50	Acct-Multi-Session-Id[1]	no	yes	yes	yes	yes	yes	yes
51	Acct-Link-Count[2]	no	yes	yes	yes	yes	yes	yes
60	CHAP-Challenge	yes	yes	yes	yes	yes	yes	yes
61	NAS-Port-Type	yes	yes	yes	yes	yes	yes	yes
62	Port-Limit	yes	yes	yes	yes	yes	yes	yes
63	Login-LAT-Port	no	no	no	no	no	no	no
64	Tunnel-Type[3]	no	no	no	no	no	no	yes
65	Tunnel-Medium-Type[3]	no	no	no	no	no	no	yes
66	Tunnel-Client-Endpoint	no	no	no	no	no	no	yes
67	Tunnel-Server-Endpoint[3]	no	no	no	no	no	no	yes
69	Tunnel-Password[3]	no	no	no	no	no	no	yes
82	Tunnel-Assignment-ID[3]	no	no	no	no	no	no	yes
85	Acct-Interim-Interval	no	no	no	no	no	no	yes
200	IETF-Token-Immediate	no	no	no	no	no	no	no

[1] Only stop records contain multi-session IDs. This is because start records are issued before any multilink processing takes place.

[2] Only stop records contain link counts. This is because start records are issued before any multilink processing takes place.

[3] This RADIUS attribute complies with the following two draft IETF documents: "RADIUS Attributes for Tunnel Protocol Support" and "RADIUS Accounting Modifications for Tunnel Protocol Support."

Table 32 lists Cisco-supported vendor-proprietary RADIUS attributes and the Cisco IOS release in which they are implemented. In cases where the attribute has a security server-specific format, the format is specified.

Note Attributes implemented in special (AA) or early development (T) releases will be added to the next mainline image.

Table 32 Supported Vendor-Proprietary RADIUS Attributes

Number	Vendor-Proprietary Attribute	11.1	11.2	11.3	11.3AA	11.3T	12.0	12.1
17	Change-Password	no	no	yes	yes	yes	yes	yes
21	Password-Expiration	no	no	yes	yes	yes	yes	yes
68	Tunnel-ID	no	no	no	no	no	no	no
108	My-Endpoint-Disc-Alias	no	no	no	no	no	no	no
109	My-Name-Alias	no	no	no	no	no	no	no
110	Remote-FW	no	no	no	no	no	no	no
111	Multicast-GLeave-Delay	no	no	no	no	no	no	no

Table 32 Supported Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	11.1	11.2	11.3	11.3AA	11.3T	12.0	12.1
112	CBCP-Enable	no	no	no	no	no	no	no
113	CBCP-Mode	no	no	no	no	no	no	no
114	CBCP-Delay	no	no	no	no	no	no	no
115	CBCP-Trunk-Group	no	no	no	no	no	no	no
116	Appletalk-Route	no	no	no	no	no	no	no
117	Appletalk-Peer-Mode	no	no	no	no	no	no	no
118	Route-Appletalk	no	no	no	no	no	no	no
119	FCP-Parameter	no	no	no	no	no	no	no
120	Modem-PortNo	no	no	no	no	no	no	no
121	Modem-SlotNo	no	no	no	no	no	no	no
122	Modem-ShelfNo	no	no	no	no	no	no	no
123	Call-Attempt-Limit	no	no	no	no	no	no	no
124	Call-Block-Duration	no	no	no	no	no	no	no
125	Maximum-Call-Duration	no	no	no	no	no	no	no
126	Router-Preference	no	no	no	no	no	no	no
127	Tunneling-Protocol	no	no	no	no	no	no	no
128	Shared-Profile-Enable	no	no	no	no	no	no	no
129	Primary-Home-Agent	no	no	no	no	no	no	no
130	Secondary-Home-Agent	no	no	no	no	no	no	no
131	Dialout-Allowed	no	no	no	no	no	no	no
133	BACP-Enable	no	no	no	no	no	no	no
134	DHCP-Maximum-Leases	no	no	no	no	no	no	no
135	Primary-DNS-Server	no	no	no	no	yes	yes	yes
136	Secondary-DNS-Server	no	no	no	no	yes	yes	yes
137	Client-Assign-DNS	no	no	no	no	no	no	no
138	User-Acct-Type	no	no	no	no	no	no	no
139	User-Acct-Host	no	no	no	no	no	no	no
140	User-Acct-Port	no	no	no	no	no	no	no
141	User-Acct-Key	no	no	no	no	no	no	no
142	User-Acct-Base	no	no	no	no	no	no	no
143	User-Acct-Time	no	no	no	no	no	no	no
144	Assign-IP-Client	no	no	no	no	no	no	no
145	Assign-IP-Server	no	no	no	no	no	no	no
146	Assign-IP-Global-Pool	no	no	no	no	no	no	no
147	DHCP-Reply	no	no	no	no	no	no	no
148	DHCP-Pool-Number	no	no	no	no	no	no	no
149	Expect-Callback	no	no	no	no	no	no	no

Table 32 Supported Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	11.1	11.2	11.3	11.3AA	11.3T	12.0	12.1
150	Event-Type	no	no	no	no	no	no	no
151	Session-Svr-Key	no	no	no	yes	no	no	yes
152	Multicast-Rate-Limit	no	no	no	yes	no	no	yes
153	IF-Netmask	no	no	no	no	no	no	no
154	Remote-Addr	no	no	no	no	no	no	no
155	Multicast-Client	no	no	no	yes	no	no	yes
156	FR-Circuit-Name	no	no	no	no	no	no	no
157	FR-LinkUp	no	no	no	no	no	no	no
158	FR-Nailed-Grp	no	no	no	no	no	no	no
159	FR-Type	no	no	no	no	no	no	no
160	FR-Link-Mgt	no	no	no	no	no	no	no
161	FR-N391	no	no	no	no	no	no	no
162	FR-DCE-N392	no	no	no	no	no	no	no
163	FR-DTE-N392	no	no	no	no	no	no	no
164	FR-DCE-N393	no	no	no	no	no	no	no
165	FR-DTE-N393	no	no	no	no	no	no	no
166	FR-T391	no	no	no	no	no	no	no
167	FR-T392	no	no	no	no	no	no	no
168	Bridge-Address	no	no	no	no	no	no	no
169	TS-Idle-Limit	no	no	no	no	no	no	no
170	TS-Idle-Mode	no	no	no	no	no	no	no
171	DBA-Monitor	no	no	no	no	no	no	no
172	Base-Channel-Count	no	no	no	no	no	no	no
173	Minimum-Channels	no	no	no	no	no	no	no
174	IPX-Route	no	no	no	no	no	no	no
175	FT1-Caller	no	no	no	no	no	no	no
176	Backup	no	no	no	no	no	no	no
177	Call-Type	no	no	no	no	no	no	no
178	Group	no	no	no	no	no	no	no
179	FR-DLCI	no	no	no	no	no	no	no
180	FR-Profile-Name	no	no	no	no	no	no	no
181	Ara-PW	no	no	no	no	no	no	no
182	IPX-Node-Addr	no	no	no	no	no	no	no
183	Home-Agent-IP-Addr	no	no	no	no	no	no	no
184	Home-Agent-Password	no	no	no	no	no	no	no
185	Home-Network-Name	no	no	no	no	no	no	no
186	Home-Agent-UDP-Port	no	no	no	no	no	no	no

Table 32 Supported Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	11.1	11.2	11.3	11.3AA	11.3T	12.0	12.1
187	Multilink-ID	no	no	no	yes	yes	yes	yes
188	Num-In-Multilink	no	no	no	yes	yes	yes	yes
189	First-Dest	no	no	no	no	no	no	no
190	Pre-Input-Octets	no	no	no	yes	yes	yes	yes
191	Pre-Output-Octets	no	no	no	yes	yes	yes	yes
192	Pre-Input-Packets	no	no	no	yes	yes	yes	yes
193	Pre-Output-Packets	no	no	no	yes	yes	yes	yes
194	Maximum-Time	no	no	yes	yes	yes	yes	yes
195	Disconnect-Cause	no	no	yes	yes	yes	yes	yes
196	Connect-Progress	no	no	no	no	no	no	yes
197	Data-Rate	no	no	no	no	yes	yes	yes
198	PreSession-Time	no	no	no	yes	yes	yes	yes
199	Token-Idle	no	no	no	no	no	no	no
201	Require-Auth	no	no	no	no	no	no	no
202	Number-Sessions	no	no	no	no	no	no	no
203	Authen-Alias	no	no	no	no	no	no	no
204	Token-Expiry	no	no	no	no	no	no	no
205	Menu-Selector	no	no	no	no	no	no	no
206	Menu-Item	no	no	no	no	no	no	no
207	PW-Warntime	no	no	no	no	no	no	no
208	PW-Lifetime	no	no	yes	yes	yes	yes	yes
209	IP-Direct	no	no	no	no	no	no	no
210	PPP-VJ-Slot-Comp	no	no	yes	yes	yes	yes	yes
211	PPP-VJ-1172	no	no	no	no	no	no	no
212	PPP-Async-Map	no	no	no	no	no	no	no
213	Third-Prompt	no	no	no	no	no	no	no
214	Send-Secret	no	no	no	no	no	no	yes
215	Receive-Secret	no	no	no	no	no	no	no
216	IPX-Peer-Mode	no	no	no	no	no	no	no
217	IP-Pool-Definition	no	no	yes	yes	yes	yes	yes
218	Assign-IP-Pool	no	no	yes	yes	yes	yes	yes
219	FR-Direct	no	no	no	no	no	no	no
220	FR-Direct-Profile	no	no	no	no	no	no	no
221	FR-Direct-DLCI	no	no	no	no	no	no	no
222	Handle-IPX	no	no	no	no	no	no	no
223	Netware-Timeout	no	no	no	no	no	no	no
224	IPX-Alias	no	no	no	no	no	no	no

Table 32 Supported Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	11.1	11.2	11.3	11.3AA	11.3T	12.0	12.1
225	Metric	no	no	no	no	no	no	no
226	PRI-Number-Type	no	no	no	no	no	no	no
227	Dial-Number	no	no	no	no	no	no	yes
228	Route-IP	no	no	yes	yes	yes	yes	yes
229	Route-IPX	no	no	no	no	no	no	no
230	Bridge	no	no	no	no	no	no	no
231	Send-Auth	no	no	no	no	no	no	yes
232	Send-Passwd	no	no	no	no	no	no	no
233	Link-Compression	no	no	yes	yes	yes	yes	yes
234	Target-Util	no	no	no	yes	no	yes	yes
235	Maximum-Channels	no	no	yes	yes	yes	yes	yes
236	Inc-Channel-Count	no	no	no	no	no	no	no
237	Dec-Channel-Count	no	no	no	no	no	no	no
238	Seconds-of-History	no	no	no	no	no	no	no
239	History-Weigh-Type	no	no	no	no	no	no	no
240	Add-Seconds	no	no	no	no	no	no	no
241	Remove-Seconds	no	no	no	no	no	no	no
242	Data-Filter	no	no	yes	yes	yes	yes	yes
243	Call-Filter	no	no	no	no	no	no	no
244	Idle-Limit	no	no	yes	yes	yes	yes	yes
245	Preempt-Limit	no	no	no	no	no	no	no
246	Callback	no	no	no	no	no	no	no
247	Data-Svc	no	no	no	no	no	no	yes
248	Force-56	no	no	no	no	no	no	yes
249	Billing Number	no	no	no	no	no	no	no
250	Call-By-Call	no	no	no	no	no	no	no
251	Transit-Number	no	no	no	no	no	no	no
252	Host-Info	no	no	no	no	no	no	no
253	PPP-Address	no	no	no	no	no	no	no
254	MPP-Idle-Percent	no	no	no	no	no	no	no
255	Xmit-Rate	no	no	no	yes	yes	yes	yes

For more information about Cisco's implementation of RADIUS, refer to the "Configuring RADIUS" chapter.

Comprehensive List of RADIUS Attributes

The following two sections provide a comprehensive listing and description of known RADIUS attributes:

- RADIUS IETF Attributes
- RADIUS Vendor-Proprietary Attributes

RADIUS IETF Attributes

Table 33 lists and describes IETF RADIUS attributes. In cases where the attribute has a security server-specific format, the format is specified.

Table 33 RADIUS IETF Attributes

Number	IETF Attribute	Description
1	User-Name	Indicates the name of the user being authenticated.
2	User-Password	Indicates the user's password or the user's input following an Access-Challenge. Passwords longer than 16 characters are encrypted using the IETF Draft #2 (or later) specifications.
3	CHAP-Password	Indicates the response value provided by a PPP Challenge-Handshake Authentication Protocol (CHAP) user in response to an Access-Challenge.
4	NAS-IP Address	Specifies the IP address of the network access server that is requesting authentication.
5	NAS-Port	Indicates the physical port number of the network access server that is authenticating the user. The NAS-Port value (32 bits) consists of one or two 16-bit values (depending on the setting of the **radius-server extended-portnames** command). Each 16-bit number should be viewed as a 5-digit decimal integer for interpretation as follows: For asynchronous terminal lines, async network interfaces, and virtual async interfaces, the value is **00ttt**, where **ttt** is the line number or async interface unit number. For ordinary synchronous network interface, the value is **10xxx**. For channels on a primary rate ISDN interface, the value is **2ppcc**. For channels on a basic rate ISDN interface, the value is **3bb0c**. For other types of interfaces, the value is **6nnss**.

Table 33 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
6	Service-Type	Indicates the type of service requested or the type of service to be provided. • In a request: Framed for known PPP or SLIP connection. Administrative-user for **enable** command. • In response: Login—Make a connection. Framed—Start SLIP or PPP. Administrative User—Start an EXEC or **enable ok**. Exec User—Start an EXEC session. Service type is indicated by a particular numeric value as follows: • 1: Login • 2: Framed • 3: Callback-Login • 4: Callback-Framed • 5: Outbound • 6: Administrative • 7: NAS-Prompt • 8: Authenticate Only • 9: Callback-NAS-Prompt
7	Framed-Protocol	Indicates the framing to be used for framed access. Framing is indicated by a numeric value as follows: • 1: PPP • 2: SLIP • 3: ARA • 4: Gandalf-proprietary single-link/multilink protocol • 5: Xylogics-proprietary IPX/SLIP
8	Framed-IP-Address	Indicates the IP address to be configured for the user.
9	Framed-IP-Netmask	Indicates the IP netmask to be configured for the user when the user is a router to a network. This attribute value results in a static route being added for Framed-IP-Address with the mask specified.
10	Framed-Routing	Indicates the routing method for the user when the user is a router to a network. Only "None" and "Send and Listen" values are supported for this attribute. Routing method is indicated by a numeric value as follows: • 0: None • 1: Send routing packets • 2: Listen for routing packets • 3: Send routing packets and listen for routing packets

Table 33 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
11	Filter-Id	Indicates the name of the filter list for the user and is formatted as follows: %d, %d.in, or %d.out. This attribute is associated with the most recent service-type command. For login and EXEC, use %d or %d.out as the line access list value from 0 to 199. For Framed service, use %d or %d.out as interface output access list, and %d.in for input access list. The numbers are self-encoding to the protocol to which they refer.
12	Framed-MTU	Indicates the maximum transmission unit (MTU) that can be configured for the user when the MTU is not negotiated by PPP or some other means.
13	Framed-Compression	Indicates a compression protocol used for the link. This attribute results in a "/compress" being added to the PPP or SLIP autocommand generated during EXEC authorization. Not currently implemented for non-EXEC authorization. Compression protocol is indicated by a numeric value as follows: • 0: None • 1: VJ-TCP/IP header compression • 2: IPX header compression
14	Login-IP-Host	Indicates the host to which the user will connect when the Login-Service attribute is included.
15	Login-Service	Indicates the service that should be used to connect the user to the login host. Service is indicated by a numeric value as follows: • 0: Telnet • 1: Rlogin • 2: TCP-Clear • 3: PortMaster • 4: LAT
16	Login-TCP-Port	Defines the TCP port with which the user is to be connected when the Login-Service attribute is also present.
18	Reply-Message	Indicates text that might be displayed to the user.
19	Callback-Number	Defines a dialing string to be used for callback.
20	Callback-ID	Defines the name (consisting of one or more octets) of a place to be called, to be interpreted by the network access server.
22	Framed-Route	Provides routing information to be configured for the user on this network access server. The RADIUS RFC format (net/bits [router [metric]]) and the old style dotted mask (net mask [router [metric]]) are supported. If the router field is omitted or 0, the peer IP address is used. Metrics are currently ignored.
23	Framed-IPX-Network	Defines the IPX network number configured for the user.
24	State	Allows state information to be maintained between the network access server and the RADIUS server. This attribute is applicable only to CHAP challenges.
25	Class	(Accounting) Arbitrary value that the network access server includes in all accounting packets for this user if supplied by the RADIUS server.

Table 33 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description

26	Vendor-Specific	Allows vendors to support their own extended attributes not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. Cisco's vendor-ID is 9, and the supported option has vendor-type 1, which is named "cisco-avpair." The value is a string of the format:
		`protocol : attribute sep value`
		"Protocol" is a value of the Cisco "protocol" attribute for a particular type of authorization. "Attribute" and "value" are an appropriate AV pair defined in the Cisco TACACS+ specification, and "sep" is "=" for mandatory attributes and "*" for optional attributes. This allows the full set of features available for TACACS+ authorization to also be used for RADIUS. For example:
		`cisco-avpair= "ip:addr-pool=first"`
		`cisco-avpair= "shell:priv-lvl=15"`
		The first example causes Cisco's "multiple named ip address pools" feature to be activated during IP authorization (during PPP's IPCP address assignment). The second example causes a user logging in from a network access server to have immediate access to EXEC commands.
		Table 34 lists supported vendor-specific RADIUS attributes (IETF attribute 26). The "TACACS+ Attribute-Value Pairs" appendix provides a complete list of supported TACACS+ attribute-value (AV) pairs that can be used with IETF attribute 26.

27	Session-Timeout	Sets the maximum number of seconds of service to be provided to the user before the session terminates. This attribute value becomes the per-user "absolute timeout."
28	Idle-Timeout	Sets the maximum number of consecutive seconds of idle connection allowed to the user before the session terminates. This attribute value becomes the per-user "session-timeout."
29	Termination-Action	Termination is indicated by a numeric value as follows: • 0: Default • 1: RADIUS request
30	Called-Station-Id	(Accounting) Allows the network access server to send the telephone number the user called as part of the Access-Request packet (using Dialed Number Identification [DNIS] or similar technology). This attribute is only supported on ISDN, and modem calls on the Cisco AS5200 if used with PRI.
31	Calling-Station-Id	(Accounting) Allows the network access server to send the telephone number the call came from as part of the Access-Request packet (using Automatic Number Identification or similar technology). This attribute has the same value as "remote-addr" from TACACS+. This attribute is only supported on ISDN, and modem calls on the Cisco AS5200 if used with PRI.
32	NAS-Identifier	String identifying the network access server originating the Access-Request.
33	Proxy-State	Attribute that can be sent by a proxy server to another server when forwarding Access-Requests; this must be returned unmodified in the Access-Accept, Access-Reject or Access-Challenge and removed by the proxy server before sending the response to the network access server.

Table 33 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
34	Login-LAT-Service	Indicates the system with which the user is to be connected by LAT. This attribute is only available in the EXEC mode.
35	Login-LAT-Node	Indicates the node with which the user is to be automatically connected by LAT.
36	Login-LAT-Group	Identifies the LAT group codes that this user is authorized to use.
37	Framed-AppleTalk-Link	Indicates the AppleTalk network number that should be used for serial links to the user, which is another AppleTalk router.
38	Framed-AppleTalk-Network	Indicates the AppleTalk network number that the network access server uses to allocate an AppleTalk node for the user.
39	Framed-AppleTalk-Zone	Indicates the AppleTalk Default Zone to be used for this user.
40	Acct-Status-Type	(Accounting) Indicates whether this Accounting-Request marks the beginning of the user service (start) or the end (stop).
41	Acct-Delay-Time	(Accounting) Indicates how many seconds the client has been trying to send a particular record.
42	Acct-Input-Octets	(Accounting) Indicates how many octets have been received from the port over the course of this service being provided.
43	Acct-Output-Octets	(Accounting) Indicates how many octets have been sent to the port in the course of delivering this service.
44	Acct-Session-Id	(Accounting) A unique accounting identifier that makes it easy to match start and stop records in a log file. Acct-Session ID numbers restart at 1 each time the router is power cycled or the software is reloaded.
45	Acct-Authentic	(Accounting) Indicates how the user was authenticated, whether by RADIUS, the network access server itself, or another remote authentication protocol. This attribute is set to "radius" for users authenticated by RADIUS; "remote" for TACACS+ and Kerberos; or "local" for local, enable, line, and if-needed methods. For all other methods, the attribute is omitted.
46	Acct-Session-Time	(Accounting) Indicates how long (in seconds) the user has received service.
47	Acct-Input-Packets	(Accounting) Indicates how many packets have been received from the port over the course of this service being provided to a framed user.
48	Acct-Output-Packets	(Accounting) Indicates how many packets have been sent to the port in the course of delivering this service to a framed user.

Table 33 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
49	Acct-Terminate-Cause	(Accounting) Reports details on why the connection was terminated. Termination causes are indicated by a numeric value as follows: **1** User request **2** Lost carrier **3** Lost service **4** Idle timeout **5** Session timeout **6** Admin reset **7** Admin reboot **8** Port error **9** NAS error **10** NAS request **11** NAS reboot **12** Port unneeded **13** Port pre-empted **14** Port suspended **15** Service unavailable **16** Callback **17** User error **18** Host request **Note** For attribute 49, Cisco IOS supports values 1 to 6, 9, 12, and 15 to 18.
50	Acct-Multi-Session-Id[1]	(Accounting) A unique accounting identifier used to link multiple related sessions in a log file. Each linked session in a multilink session has a unique Acct-Session-Id value, but shares the same Acct-Multi-Session-Id.
51	Acct-Link-Count[2]	(Accounting) Indicates the number of links known in a given multilink session at the time an accounting record is generated. The network access server can include this attribute in any accounting request that might have multiple links.
60	CHAP-Challenge	Contains the Challenge Handshake Authentication Protocol challenge sent by the network access server to a PPP CHAP user.
61	NAS-Port-Type	Indicates the type of physical port the network access server is using to authenticate the user. Physical ports are indicated by a numeric value as follows: • 0: Asynchronous • 1: Synchronous • 2: ISDN-Synchronous • 3: ISDN-Asynchronous (V.120) • 4: ISDN-Asynchronous (V.110) • 5: Virtual
62	Port-Limit	Sets the maximum number of ports provided to the user by the NAS.
63	Login-LAT-Port	Defines the port with which the user is to be connected by LAT.

Table 33 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
64	Tunnel-Type[3]	Indicates the tunneling protocol(s) used. Cisco IOS software supports two possible values for this attribute: L2TP and L2F. If this attribute is not set, L2F is used as a default.
65	Tunnel-Medium-Type[3]	Indicates the transport medium type to use to create a tunnel. This attribute only has one available value for this release: IP. If no value is set for this attribute, IP is used as the default.
66	Tunnel-Client-Endpoint	Contains the address of the initiator end of the tunnel. It *may* be included in both Access-Request and Access-Accept packets to indicate the address from which a new tunnel is to be initiated. If the Tunnel-Client-Endpoint attribute is included in an Access-Request packet, the RADIUS server should take the value as a hint; the server is not obligated to honor the hint, however. This attribute *should* be included in Accounting-Request packets that contain Acct-Status-Type attributes with values of either Start or Stop, in which case it indicates the address from which the tunnel was initiated. This attribute, along with the Tunnel-Server-Endpoint and Acct-Tunnel-Connection-ID attributes, may be used to provide a globally unique means to identify a tunnel for accounting and auditing purposes. An enhancement has been added for the network access server to accept a value of 127.0.0.X for this attribute such that: 127.0.0.0 would indicate that loopback0 IP address is to be used 127.0.0.1 would indicate that loopback1 IP address is to be used ... 127.0.0.X would indicate that loopbackX IP address is to be used for the actual tunnel client endpoint IP address. This enhancement adds scalability across multiple network access servers.
67	Tunnel-Server-Endpoint[3]	Indicates the address of the server end of the tunnel. The format of this attribute varies depending on the value of Tunnel-Medium-Type. Because this release only supports IP as a tunnel medium type, the IP address or the host name of LNS is valid for this attribute.
69	Tunnel-Password[3]	Defines the password to be used to authenticate to a remote server. This attribute is converted into different AAA attributes based on the value of Tunnel-Type: AAA_ATTR_l2tp_tunnel_pw (L2TP), AAA_ATTR_nas_password (L2F), and AAA_ATTR_gw_password (L2F).
82	Tunnel-Assignment-ID[3]	Indicates to the tunnel initiator the particular tunnel to which a session is assigned.
85	Acct-Interim-Interval	Indicates the number of seconds between each interim update in seconds for this specific session. This value can only appear in the Access-Accept message.
200	IETF-Token-Immediate	Determines how RADIUS treats passwords received from login-users when their file entry specifies a hand-held security card server. The value for this attribute is indicated by a numeric value as follows: • 0: No, meaning that the password is ignored. • 1: Yes, meaning that the password is used for authentication.

[1] Only stop records contain multi-session IDs. This is because start records are issued before any multilink processing takes place.

[2] Only stop records contain link counts. This is because start records are issued before any multilink processing takes place.

[3] This RADIUS attribute complies with the following two IETF documents: "RADIUS Attributes for Tunnel Protocol Support" and "RADIUS Accounting Modifications for Tunnel Protocol Support."

Table 34 lists supported vendor-specific RADIUS attributes (IETF attribute 26).

Table 34 Vendor-Specific RADIUS IETF Attributes

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
			MS-CHAP Attributes	
26	311	1	MSCHAP-Response	Contains the response value provided by a PPP MS-CHAP user in response to the challenge. It is only used in Access-Request packets. This attribute is identical to the PPP CHAP Identifier.
26	311	11	MSCHAP-Challenge	Contains the challenge sent by a network access server to an MS-CHAP user. It can be used in both Access-Request and Access-Challenge packets.
			VPDN Attributes	
26	9	1	l2tp-busy-disconnect	If a vpdn-group on an LNS uses a virtual-template that is configured to be pre-cloned, this attribute will control the disposition of a new L2TP session that finds no pre-cloned interface to which to connect. If the attribute is true (the default), the session will be disconnected by the LNS. Otherwise, a new interface will be cloned from the virtual-template.
26	9	1	l2tp-cm-local-window-size	Specifies the maximum receive window size for L2TP control messages. This value is advertised to the peer during tunnel establishment.
26	9	1	l2tp-drop-out-of-order	Respects sequence numbers on data packets by dropping those that are received out of order. This does not ensure that sequence numbers will be sent on data packets, just how to handle them if they are received.
26	9	1	l2tp-hello-interval	Specifies the number of seconds for the hello keepalive interval. Hello packets are sent when no data has been sent on a tunnel for the number of seconds configured here.
26	9	1	l2tp-hidden-avp	When enabled, sensitive AVPs in L2TP control messages are scrambled or hidden.

Table 34 Vendor-Specific RADIUS IETF Attributes (continued)

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	1	l2tp-nosession-timeout	Specifies the number of seconds that a tunnel will stay active with no sessions before timing out and shutting down.
26	9	1	l2tp-tos-reflect	Copies the IP ToS field from the IP header of each payload packet to the IP header of the tunnel packet for packets entering the tunnel at the LNS.
26	9	1	l2tp-tunnel-authen	If this attribute is set, it performs L2TP tunnel authentication.
26	9	1	l2tp-tunnel-password	Shared secret used for L2TP tunnel authentication and AVP hiding.
26	9	1	l2tp-udp-checksum	This is an authorization attribute and defines whether L2TP should perform UDP checksums for data packets. Valid values are "yes" and "no." The default is no.
		Store and Forward Fax Attributes		
26	9	3	Fax-Account-Id-Origin	Indicates the account ID origin as defined by system administrator for the **mmoip aaa receive-id** or the **mmoip aaa send-id** commands.
26	9	4	Fax-Msg-Id=	Indicates a unique fax message identification number assigned by Store and Forward Fax.
26	9	5	Fax-Pages	Indicates the number of pages transmitted or received during this fax session. This page count includes cover pages.
26	9	6	Fax-Coverpage-Flag	Indicates whether or not a cover page was generated by the off-ramp gateway for this fax session. True indicates that a cover page was generated; false means that a cover page was not generated.
26	9	7	Fax-Modem-Time	Indicates the amount of time in seconds the modem sent fax data (x) and the amount of time in seconds of the total fax session (y), which includes both fax-mail and PSTN time, in the form x/y. For example, 10/15 means that the transfer time took 10 seconds, and the total fax session took 15 seconds.

Table 34 Vendor-Specific RADIUS IETF Attributes (continued)

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	8	Fax-Connect-Speed	Indicates the modem speed at which this fax-mail was initially transmitted or received. Possible values are 1200, 4800, 9600, and 14400.
26	9	9	Fax-Recipient-Count	Indicates the number of recipients for this fax transmission. Until e-mail servers support Session mode, the number should be 1.
26	9	10	Fax-Process-Abort-Flag	Indicates that the fax session was aborted or successful. True means that the session was aborted; false means that the session was successful.
26	9	11	Fax-Dsn-Address	Indicates the address to which DSNs will be sent.
26	9	12	Fax-Dsn-Flag	Indicates whether or not DSN has been enabled. True indicates that DSN has been enabled; false means that DSN has not been enabled.
26	9	13	Fax-Mdn-Address	Indicates the address to which MDNs will be sent.
26	9	14	Fax-Mdn-Flag	Indicates whether or not message delivery notification (MDN) has been enabled. True indicates that MDN had been enabled; false means that MDN had not been enabled.
26	9	15	Fax-Auth-Status	Indicates whether or not authentication for this fax session was successful. Possible values for this field are success, failed, bypassed, or unknown.
26	9	16	Email-Server-Address	Indicates the IP address of the e-mail server handling the on-ramp fax-mail message.
26	9	17	Email-Server-Ack-Flag	Indicates that the on-ramp gateway has received a positive acknowledgment from the e-mail server accepting the fax-mail message.
26	9	18	Gateway-Id	Indicates the name of the gateway that processed the fax session. The name appears in the following format: hostname.domain-name.
26	9	19	Call-Type	Describes the type of fax activity: fax receive or fax send.

Table 34 Vendor-Specific RADIUS IETF Attributes (continued)

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	20	Port-Used	Indicates the slot/port number of the Cisco AS5300 used to either transmit or receive this fax-mail.
26	9	21	Abort-Cause	If the fax session aborts, indicates the system component that signaled the abort. Examples of system components that could trigger an abort are FAP (Fax Application Process), TIFF (the TIFF reader or the TIFF writer), fax-mail client, fax-mail server, ESMTP client, or ESMTP server.
			H323 Attributes	
26	9	23	h323-remote-address	Indicates the IP address of the remote gateway.
26	9	24	h323-conf-id	Identifies the conference ID.
26	9	25	h323-setup-time	Indicates the setup time for this connection in Coordinated Universal Time (UTC) formerly known as Greenwich Mean Time (GMT) and Zulu time.
26	9	26	h323-call-origin	Indicates the origin of the call relative to the gateway. Possible values are originating and terminating (answer).
26	9	27	h323-call-type	Indicates call leg type. Possible values are **telephony** and **VoIP**.
26	9	28	h323-connect-time	Indicates the connection time for this call leg in UTC.
26	9	29	h323-disconnect-time	Indicates the time this call leg was disconnected in UTC.
26	9	30	h323-disconnect-cause	Specifies the reason a connection was taken offline per Q.931 specification.
26	9	31	h323-voice-quality	Specifies the impairment factor (ICPIF) affecting voice quality for a call.
26	9	33	h323-gw-id	Indicates the name of the underlying gateway.
			Large Scale Dialout Attributes	
26	9	1	callback-dialstring	Defines a dialing string to be used for callback.
26	9	1	data-service	No description available.
26	9	1	dial-number	Defines the number to dial.

Table 34 Vendor-Specific RADIUS IETF Attributes (continued)

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	1	force-56	Determines whether the network access server uses only the 56 K portion of a channel, even when all 64 K appear to be available.
26	9	1	map-class	Allows the user profile to reference information configured in a map class of the same name on the network access server that dials out.
26	9	1	send-auth	Defines the protocol to use (PAP or CHAP) for username-password authentication following CLID authentication.
			Miscellaneous Attributes	
26	9	2	Cisco-NAS-Port	Specifies additional vendor specific attribute (VSA) information for NAS-Port accounting.
26	9	1	min-links	Sets the minimum number of links for MLP.
26	9	1	proxyacl#<n>	Allows users to configure the downloadable user profiles (dynamic ACLs) by using the authentication proxy feature so that users can have the configured authorization to permit traffic going through the configured interfaces.
26	9	1	spi	Carries the authentication information needed by the home agent to authenticate a mobile node during registration. The information is in the same syntax as the **ip mobile secure host <addr>** configuration command. Basically it contains the rest of the configuration command that follows that string, verbatim. It provides the Security Parameter Index (SPI), key, authentication algorithm, authentication mode, and replay protection timestamp range.

RADIUS Vendor-Proprietary Attributes

Although an Internet Engineering Task Force (IETF) draft standard for RADIUS specifies a method for communicating vendor-proprietary information between the network access server and the RADIUS server, some vendors have extended the RADIUS attribute set in a unique way. Table 35 lists the known vendor-proprietary RADIUS attributes:

Table 35 Vendor-Proprietary RADIUS Attributes

Number	Vendor-Proprietary Attribute	Description
17	Change-Password	Specifies a request to change a user's password.
21	Password-Expiration	Specifies an expiration date for a user's password in the user's file entry.
68	Tunnel-ID	(Ascend 5) No description available.
108	My-Endpoint-Disc-Alias	(Ascend 5) No description available.
109	My-Name-Alias	(Ascend 5) No description available.
110	Remote-FW	(Ascend 5) No description available.
111	Multicast-GLeave-Delay	(Ascend 5) No description available.
112	CBCP-Enable	(Ascend 5) No description available.
113	CBCP-Mode	(Ascend 5) No description available.
114	CBCP-Delay	(Ascend 5) No description available.
115	CBCP-Trunk-Group	(Ascend 5) No description available.
116	Appletalk-Route	(Ascend 5) No description available.
117	Appletalk-Peer-Mode	(Ascend 5) No description available.
118	Route-Appletalk	(Ascend 5) No description available.
119	FCP-Parameter	(Ascend 5) No description available.
120	Modem-PortNo	(Ascend 5) No description available.
121	Modem-SlotNo	(Ascend 5) No description available.
122	Modem-ShelfNo	(Ascend 5) No description available.
123	Call-Attempt-Limit	(Ascend 5) No description available.
124	Call-Block-Duration	(Ascend 5) No description available.
125	Maximum-Call-Duration	(Ascend 5) No description available.
126	Router-Preference	(Ascend 5) No description available.
127	Tunneling-Protocol	(Ascend 5) No description available.
128	Shared-Profile-Enable	(Ascend 5) No description available.
129	Primary-Home-Agent	(Ascend 5) No description available.
130	Secondary-Home-Agent	(Ascend 5) No description available.
131	Dialout-Allowed	(Ascend 5) No description available.
133	BACP-Enable	(Ascend 5) No description available.
134	DHCP-Maximum-Leases	(Ascend 5) No description available.
135	Primary-DNS-Server	Identifies a primary DNS server that can be requested by Microsoft PPP clients from the network access server during IPCP negotiation.

Table 35 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Description
136	Secondary-DNS-Server	Identifies a secondary DNS server that can be requested by Microsoft PPP clients from the network access server during IPCP negotiation.
137	Client-Assign-DNS	No description available.
138	User-Acct-Type	No description available.
139	User-Acct-Host	No description available.
140	User-Acct-Port	No description available.
141	User-Acct-Key	No description available.
142	User-Acct-Base	No description available.
143	User-Acct-Time	No description available.
144	Assign-IP-Client	No description available.
145	Assign-IP-Server	No description available.
146	Assign-IP-Global-Pool	No description available.
147	DHCP-Reply	No description available.
148	DHCP-Pool-Number	No description available.
149	Expect-Callback	No description available.
150	Event-Type	No description available.
151	Session-Svr-Key	No description available.
152	Multicast-Rate-Limit	No description available.
153	IF-Netmask	No description available.
154	Remote-Addr	No description available.
155	Multicast-Client	No description available.
156	FR-Circuit-Name	No description available.
157	FR-LinkUp	No description available.
158	FR-Nailed-Grp	No description available.
159	FR-Type	No description available.
160	FR-Link-Mgt	No description available.
161	FR-N391	No description available.
162	FR-DCE-N392	No description available.
163	FR-DTE-N392	No description available.
164	FR-DCE-N393	No description available.
165	FR-DTE-N393	No description available.
166	FR-T391	No description available.
167	FR-T392	No description available.
168	Bridge-Address	No description available.
169	TS-Idle-Limit	No description available.
170	TS-Idle-Mode	No description available.
171	DBA-Monitor	No description available.
172	Base-Channel-Count	No description available.

Table 35 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Description
173	Minimum-Channels	No description available.
174	IPX-Route	No description available.
175	FT1-Caller	No description available.
176	Backup	No description available.
177	Call-Type	No description available.
178	Group	No description available.
179	FR-DLCI	No description available.
180	FR-Profile-Name	No description available.
181	Ara-PW	No description available.
182	IPX-Node-Addr	No description available.
183	Home-Agent-IP-Addr	Indicates the home agent's IP address (in dotted decimal format) when using Ascend Tunnel Management Protocol (ATMP).
184	Home-Agent-Password	With ATMP, specifies the password that the foreign agent uses to authenticate itself.
185	Home-Network-Name	With ATMP, indicates the name of the connection profile to which the home agent sends all packets.
186	Home-Agent-UDP-Port	Indicates the UDP port number the foreign agent uses to send ATMP messages to the home agent.
187	Multilink-ID	Reports the identification number of the multilink bundle when the session closes. This attribute applies to sessions that are part of a multilink bundle. The Multilink-ID attribute is sent in authentication-response packets.
188	Num-In-Multilink	Reports the number of sessions remaining in a multilink bundle when the session reported in an accounting-stop packet closes. This attribute applies to sessions that are part of a multilink bundle. The Num-In-Multilink attribute is sent in authentication-response packets and in some accounting-request packets.
189	First-Dest	Records the destination IP address of the first packet received after authentication.
190	Pre-Input-Octets	Records the number of input octets before authentication. The Pre-Input-Octets attribute is sent in accounting-stop records.
191	Pre-Output-Octets	Records the number of output octets before authentication. The Pre-Output-Octets attribute is sent in accounting-stop records.
192	Pre-Input-Packets	Records the number of input packets before authentication. The Pre-Input-Packets attribute is sent in accounting-stop records.
193	Pre-Output-Packets	Records the number of output packets before authentication. The Pre-Output-Packets attribute is sent in accounting-stop records.

Table 35 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Description
194	Maximum-Time	Specifies the maximum length of time (in seconds) allowed for any session. After the session reaches the time limit, its connection is dropped.
195	Disconnect-Cause	Specifies the reason a connection was taken offline. The Disconnect-Cause attribute is sent in accounting-stop records. This attribute also causes stop records to be generated without first generating start records if disconnection occurs before authentication is performed. Refer to Table 36 for a list of Disconnect-Cause values and their meanings.
196	Connect-Progress	Indicates the connection state before the connection is disconnected.
197	Data-Rate	Specifies the average number of bits per second over the course of the connection's lifetime. The Data-Rate attribute is sent in accounting-stop records.
198	PreSession-Time	Specifies the length of time, in seconds, from when a call first connects to when it completes authentication. The PreSession-Time attribute is sent in accounting-stop records.
199	Token-Idle	Indicates the maximum amount of time (in minutes) a cached token can remain alive between authentications.
201	Require-Auth	Defines whether additional authentication is required for class that has been CLID authenticated.
202	Number-Sessions	Specifies the number of active sessions (per class) reported to the RADIUS accounting server.
203	Authen-Alias	Defines the RADIUS server's login name during PPP authentication.
204	Token-Expiry	Defines the lifetime of a cached token.
205	Menu-Selector	Defines a string to be used to cue a user to input data.
206	Menu-Item	Specifies a single menu-item for a user-profile. Up to 20 menu items can be assigned per profile.
207	PW-Warntime	(Ascend 5) No description available.
208	PW-Lifetime	Enables you to specify on a per-user basis the number of days that a password is valid.
209	IP-Direct	Specifies in a user's file entry the IP address to which the Cisco router redirects packets from the user. When you include this attribute in a user's file entry, the Cisco router bypasses all internal routing and bridging tables and sends all packets received on this connection's WAN interface to the specified IP address.
210	PPP-VJ-Slot-Comp	Instructs the Cisco router not to use slot compression when sending VJ-compressed packets over a PPP link.
211	PPP-VJ-1172	Instructs PPP to use the 0x0037 value for VJ compression.

Table 35 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Description
212	PPP-Async-Map	Gives the Cisco router the asynchronous control character map for the PPP session. The specified control characters are passed through the PPP link as data and used by applications running over the link.
213	Third-Prompt	Defines a third prompt (after username and password) for additional user input.
214	Send-Secret	Enables an encrypted password to be used in place of a regular password in outdial profiles.
215	Receive-Secret	Enables an encrypted password to be verified by the RADIUS server.
216	IPX-Peer-Mode	(Ascend 5) No description available.
217	IP-Pool-Definition	Defines a pool of addresses using the following format: X a.b.c Z; where X is the pool index number, a.b.c is the pool's starting IP address, and Z is the number of IP addresses in the pool. For example, 3 10.0.0.1 5 allocates 10.0.0.1 through 10.0.0.5 for dynamic assignment.
218	Assign-IP-Pool	Tells the router to assign the user and IP address from the IP pool.
219	FR-Direct	Defines whether the connection profile operates in Frame Relay redirect mode.
220	FR-Direct-Profile	Defines the name of the Frame Relay profile carrying this connection to the Frame Relay switch.
221	FR-Direct-DLCI	Indicates the DLCI carrying this connection to the Frame Relay switch.
222	Handle-IPX	Indicates how NCP watchdog requests will be handled.
223	Netware-Timeout	Defines, in minutes, how long the RADIUS server responds to NCP watchdog packets.
224	IPX-Alias	Allows you to define an alias for IPX routers requiring numbered interfaces.
225	Metric	No description available.
226	PRI-Number-Type	No description available.
227	Dial-Number	Defines the number to dial.
228	Route-IP	Indicates whether IP routing is allowed for the user's file entry.
229	Route-IPX	Allows you to enable IPX routing.
230	Bridge	No description available.
231	Send-Auth	Defines the protocol to use (PAP or CHAP) for username-password authentication following CLID authentication.
232	Send-Passwd	No description available.

Table 35 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Description
233	Link-Compression	Defines whether to turn on or turn off "stac" compression over a PPP link. Link compression is defined as a numeric value as follows: • 0: None • 1: Stac • 2: Stac-Draft-9 • 3: MS-Stac
234	Target-Util	Specifies the load-threshold percentage value for bringing up an additional channel when PPP multilink is defined.
235	Maximum-Channels	Specifies allowed/allocatable maximum number of channels.
236	Inc-Channel-Count	No description available.
237	Dec-Channel-Count	No description available.
238	Seconds-of-History	No description available.
239	History-Weigh-Type	No description available.
240	Add-Seconds	No description available.
241	Remove-Seconds	No description available.
242	Data-Filter	Defines per-user IP data filters. These filters are retrieved only when a call is placed using a RADIUS outgoing profile or answered using a RADIUS incoming profile. Filter entries are applied on a first-match basis; therefore, the order in which filter entries are entered is important.
243	Call-Filter	Defines per-user IP data filters. On a Cisco router, this attribute is identical to the Data-Filter attribute.
244	Idle-Limit	Specifies the maximum time (in seconds) that any session can be idle. When the session reaches the idle time limit, its connection is dropped.
245	Preempt-Limit	No description available.
246	Callback	Allows you to enable or disable callback.
247	Data-Svc	No description available.
248	Force-56	Determines whether the network access server uses only the 56 K portion of a channel, even when all 64 K appear to be available.
249	Billing Number	No description available.
250	Call-By-Call	No description available.
251	Transit-Number	No description available.
252	Host-Info	No description available.
253	PPP-Address	Indicates the IP address reported to the calling unit during PPP IPCP negotiations.

Table 35 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Description
254	MPP-Idle-Percent	No description available.
255	Xmit-Rate	(Ascend 5) No description available.

Table 36 lists the values and descriptions for the Disconnect-Cause (195) attribute.

Table 36 Disconnect-Cause Attribute Values

Value	Description
Unknown (2)	Reason unknown.
CLID-Authentication-Failure (4)	Failure to authenticate calling-party number.
No-Carrier (10)	No carrier detected. This value applies to modem connections.
Lost-Carrier (11)	Loss of carrier. This value applies to modem connections.
No-Detected-Result-Codes (12)	Failure to detect modem result codes. This value applies to modem connections.
User-Ends-Session (20)	User terminates a session. This value applies to EXEC sessions.
Idle-Timeout (21)	Timeout waiting for user input. This value applies to all session types.
Exit-Telnet-Session (22)	Disconnect due to exiting Telnet session. This value applies to EXEC sessions.
No-Remote-IP-Addr (23)	Could not switch to SLIP/PPP; the remote end has no IP address. This value applies to EXEC sessions.
Exit-Raw-TCP (24)	Disconnect due to exiting raw TCP. This value applies to EXEC sessions.
Password-Fail (25)	Bad passwords. This value applies to EXEC sessions.
Raw-TCP-Disabled (26)	Raw TCP disabled. This value applies to EXEC sessions.
Control-C-Detected (27)	Control-C detected. This value applies to EXEC sessions.
EXEC-Process-Destroyed (28)	EXEC process destroyed. This value applies to EXEC sessions.
Timeout-PPP-LCP (40)	PPP LCP negotiation timed out. This value applies to PPP sessions.
Failed-PPP-LCP-Negotiation (41)	PPP LCP negotiation failed. This value applies to PPP sessions.
Failed-PPP-PAP-Auth-Fail (42)	PPP PAP authentication failed. This value applies to PPP sessions.
Failed-PPP-CHAP-Auth (43)	PPP CHAP authentication failed. This value applies to PPP sessions.
Failed-PPP-Remote-Auth (44)	PPP remote authentication failed. This value applies to PPP sessions.
PPP-Remote-Terminate (45)	PPP received a Terminate Request from remote end. This value applies to PPP sessions.
PPP-Closed-Event (46)	Upper layer requested that the session be closed. This value applies to PPP sessions.
Session-Timeout (100)	Session timed out. This value applies to all session types.
Session-Failed-Security (101)	Session failed for security reasons. This value applies to all session types.
Session-End-Callback (102)	Session terminated due to callback. This value applies to all session types.
Invalid-Protocol (120)	Call refused because the detected protocol is disabled. This value applies to all session types.

TACACS+ Attribute-Value Pairs

Terminal Access Controller Access Control System Plus (TACACS+) attribute-value (AV) pairs are used to define specific authentication, authorization, and accounting elements in a user profile, which is stored on the TACACS+ daemon. This appendix lists the TACACS+ AV pairs currently supported.

How to Use This Appendix

This appendix is divided into two sections:

- TACACS+ Authentication and Authorization AV Pairs
- TACACS+ Accounting AV Pairs

The first section lists and describes the supported TACACS+ authentication and authorization AV pairs, and it specifies the Cisco IOS release in which they are implemented. The second section lists and describes the supported TACACS+ accounting AV pairs, and it specifies the Cisco IOS release in which they are implemented.

TACACS+ Authentication and Authorization AV Pairs

Table 37 lists and describes the supported TACACS+ authentication and authorization AV pairs, and it specifies the Cisco IOS release in which they are implemented.

Table 37 Supported TACACS+ Authentication and Authorization AV Pairs

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1
acl=x	ASCII number representing a connection access list. Used only when service=shell.	yes	yes	yes	yes	yes	yes
addr=x	A network address. Used with service=slip, service=ppp, and protocol=ip. Contains the IP address that the remote host should use when connecting via SLIP or PPP/IP. For example, addr=10.2.3.4.	yes	yes	yes	yes	yes	yes

Table 37 Supported TACACS+ Authentication and Authorization AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1

addr-pool=x	Specifies the name of a local pool from which to get the address of the remote host. Used with service=ppp and protocol=ip.
	Note that **addr-pool** works in conjunction with local pooling. It specifies the name of a local pool (which must be preconfigured on the network access server). Use the **ip-local pool** command to declare local pools. For example:
	`ip address-pool local`
	`ip local pool boo 10.0.0.1 10.0.0.10`
	`ip local pool moo 10.0.0.1 10.0.0.20`
	You can then use TACACS+ to return addr-pool=boo or addr-pool=moo to indicate the address pool from which you want to get this remote node's address.
		yes	yes	yes	yes	yes	yes

autocmd=x	Specifies an autocommand to be executed at EXEC startup (for example, autocmd=telnet example.com). Used only with service=shell.	yes	yes	yes	yes	yes	yes
callback-dialstring	Sets the telephone number for a callback (for example: callback-dialstring= 408-555-1212). Value is NULL, or a dial-string. A NULL value indicates that the service might choose to get the dial string through other means. Used with service=arap, service=slip, service=ppp, service=shell. Not valid for ISDN.	no	yes	yes	yes	yes	yes
callback-line	The number of a TTY line to use for callback (for example: callback-line=4). Used with service=arap, service=slip, service=ppp, service=shell. Not valid for ISDN.	no	yes	yes	yes	yes	yes

Table 37 Supported TACACS+ Authentication and Authorization AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1
callback-rotary	The number of a rotary group (between 0 and 100 inclusive) to use for callback (for example: callback-rotary=34). Used with service=arap, service=slip, service=ppp, service=shell. Not valid for ISDN.	no	yes	yes	yes	yes	yes
cmd-arg=x	An argument to a shell (EXEC) command. This indicates an argument for the shell command that is to be run. Multiple cmd-arg attributes can be specified, and they are order dependent. **Note** This TACACS+ AV pair cannot be used with RADIUS attribute 26.	yes	yes	yes	yes	yes	yes
cmd=x	A shell (EXEC) command. This indicates the command name for a shell command that is to be run. This attribute must be specified if service equals "shell." A NULL value indicates that the shell itself is being referred to. **Note** This TACACS+ AV pair cannot be used with RADIUS attribute 26.	yes	yes	yes	yes	yes	yes
data-service	No description available.	no	no	no	no	no	yes
dial-number	Defines the number to dial.	no	no	no	no	no	yes
dns-servers=	Identifies a DNS server (primary or secondary) that can be requested by Microsoft PPP clients from the network access server during IPCP negotiation. To be used with service=ppp and protocol=ip. The IP address identifying each DNS server is entered in dotted decimal format.	no	no	no	yes	yes	yes
force-56	Determines whether the network access server uses only the 56 K portion of a channel, even when all 64 K appear to be available.	no	no	no	no	no	yes
gw-password	Specifies the password for the home gateway during the L2F tunnel authentication. Used with service=ppp and protocol=vpdn.	no	no	yes	yes	yes	yes

Table 37 Supported TACACS+ Authentication and Authorization AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1
idletime=x	Sets a value, in minutes, after which an idle session is terminated. Does not work for PPP. A value of zero indicates no timeout.	no	yes	yes	yes	yes	yes
inacl#<n>	ASCII access list identifier for an input access list to be installed and applied to an interface for the duration of the current connection. Used with service=ppp and protocol=ip, and service service=ppp and protocol =ipx. Per-user access lists do not currently work with ISDN interfaces.	no	no	no	yes	yes	yes
inacl=x	ASCII identifier for an interface input access list. Used with service=ppp and protocol=ip. Per-user access lists do not currently work with ISDN interfaces.	yes	yes	yes	yes	yes	yes
interface-config=	Specifies user-specific AAA interface configuration information with Virtual Profiles. The information that follows the equal sign (=) can be any Cisco IOS interface configuration command.	no	no	no	yes	yes	yes
ip-addresses	Space-separated list of possible IP addresses that can be used for the end-point of a tunnel. Used with service=ppp and protocol=vpdn.	no	no	yes	yes	yes	yes
l2tp-busy-disconnect	If a vpdn-group on an LNS uses a virtual-template that is configured to be pre-cloned, this attribute will control the disposition of a new L2TP session that finds no pre-cloned interface to which to connect. If the attribute is true (the default), the session will be disconnected by the LNS. Otherwise, a new interface will be cloned from the virtual-template.	no	no	no	no	no	yes
l2tp-cm-local-window-size	Specifies the maximum receive window size for L2TP control messages. This value is advertised to the peer during tunnel establishment.	no	no	no	no	no	yes

Table 37 Supported TACACS+ Authentication and Authorization AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1
l2tp-drop-out-of-order	Respects sequence numbers on data packets by dropping those that are received out of order. This does not ensure that sequence numbers will be sent on data packets, just how to handle them if they are received.	no	no	no	no	no	yes
l2tp-hello-interval	Specifies the number of seconds for the hello keepalive interval. Hello packets are sent when no data has been sent on a tunnel for the number of seconds configured here.	no	no	no	no	no	yes
l2tp-hidden-avp	When enabled, sensitive AVPs in L2TP control messages are scrambled or hidden.	no	no	no	no	no	yes
l2tp-nosession-timeout	Specifies the number of seconds that a tunnel will stay active with no sessions before timing out and shutting down.	no	no	no	no	no	yes
l2tp-tos-reflect	Copies the IP ToS field from the IP header of each payload packet to the IP header of the tunnel packet for packets entering the tunnel at the LNS.	no	no	no	no	no	yes
l2tp-tunnel-authen	If this attribute is set, it performs L2TP tunnel authentication.	no	no	no	no	no	yes
l2tp-tunnel-password	Shared secret used for L2TP tunnel authentication and AVP hiding.	no	no	no	no	no	yes
l2tp-udp-checksum	This is an authorization attribute and defines whether L2TP should perform UDP checksums for data packets. Valid values are "yes" and "no." The default is no.	no	no	no	no	no	yes
link-compression=	Defines whether to turn on or turn off "stac" compression over a PPP link. Link compression is defined as a numeric value as follows: • 0: None • 1: Stac • 2: Stac-Draft-9 • 3: MS-Stac	no	no	no	yes	yes	yes

Table 37 Supported TACACS+ Authentication and Authorization AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1
load-threshold=<n>	Sets the load threshold for the caller at which additional links are either added to or deleted from the multilink bundle. If the load goes above the specified value, additional links are added. If the load goes below the specified value, links are deleted. Used with service=ppp and protocol=multilink. The range for <n> is from 1 to 255.	no	no	no	yes	yes	yes
map-class	Allows the user profile to reference information configured in a map class of the same name on the network access server that dials out.	no	no	no	no	no	yes
max-links=<n>	Restricts the number of links that a user can have in a multilink bundle. Used with service=ppp and protocol=multilink. The range for <n> is from 1 to 255.	no	no	no	yes	yes	yes
min-links	Sets the minimum number of links for MLP.	no	no	no	no	no	yes
nas-password	Specifies the password for the network access server during the L2F tunnel authentication. Used with service=ppp and protocol=vpdn.	no	no	yes	yes	yes	yes
nocallback-verify	Indicates that no callback verification is required. The only valid value for this parameter is 1 (for example, nocallback-verify=1). Used with service=arap, service=slip, service=ppp, service=shell. There is no authentication on callback. Not valid for ISDN.	no	yes	yes	yes	yes	yes
noescape=x	Prevents user from using an escape character. Used with service=shell. Can be either true or false (for example, noescape=true).	yes	yes	yes	yes	yes	yes
nohangup=x	Used with service=shell. Specifies the nohangup option, which means that after an EXEC shell is terminated, the user is presented with another login (username) prompt. Can be either true or false (for example, nohangup=false).	yes	yes	yes	yes	yes	yes

Table 37 Supported TACACS+ Authentication and Authorization AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1
old-prompts	Allows providers to make the prompts in TACACS+ appear identical to those of earlier systems (TACACS and Extended TACACS). This allows administrators to upgrade from TACACS or Extended TACACS to TACACS+ transparently to users.	yes	yes	yes	yes	yes	yes
outacl#<n>	ASCII access list identifier for an interface output access list to be installed and applied to an interface for the duration of the current condition. Used with service=ppp and protocol=ip, and service service=ppp and protocol=ipx. Per-user access lists do not currently work with ISDN interfaces.	no	no	no	yes	yes	yes
outacl=x	ASCII identifier for an interface output access list. Used with service=ppp and protocol=ip, and service service=ppp and protocol=ipx. Contains an IP output access list for SLIP or PPP/IP (for example, outacl=4). The access list itself must be preconfigured on the router. Per-user access lists do not currently work with ISDN interfaces.	yes (PPP/IP only)	yes	yes	yes	yes	yes
pool-def#<n>	Defines IP address pools on the network access server. Used with service=ppp and protocol=ip.	no	no	no	yes	yes	yes
pool-timeout=	Defines (in conjunction with pool-def) IP address pools on the network access server. During IPCP address negotiation, if an IP pool name is specified for a user (see the addr-pool attribute), a check is made to see if the named pool is defined on the network access server. If it is, the pool is consulted for an IP address.	no	no	yes	yes	yes	yes

Table 37 Supported TACACS+ Authentication and Authorization AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1
port-type	Indicates the type of physical port the network access server is using to authenticate the user. Physical ports are indicated by a numeric value as follows: • 0: Asynchronous • 1: Synchronous • 2: ISDN-Synchronous • 3: ISDN-Asynchronous (V.120) • 4: ISDN- Asynchronous (V.110) • 5: Virtual	no	no	no	no	no	yes
ppp-vj-slot-compression	Instructs the Cisco router not to use slot compression when sending VJ-compressed packets over a PPP link.	no	no	no	yes	yes	yes
priv-lvl=x	Privilege level to be assigned for the EXEC. Used with service=shell. Privilege levels range from 0 to 15, with 15 being the highest.	yes	yes	yes	yes	yes	yes
protocol=x	A protocol that is a subset of a service. An example would be any PPP NCP. Currently known values are **lcp**, **ip**, **ipx**, **atalk**, **vines**, **lat**, **xremote**, **tn3270**, **telnet**, **rlogin**, **pad**, **vpdn**, **osicp**, **deccp**, **ccp**, **cdp**, **bridging**, **xns**, **nbf**, **bap**, **multilink**, and **unknown**.	yes	yes	yes	yes	yes	yes
proxyacl#<n>	Allows users to configure the downloadable user profiles (dynamic ACLs) by using the authentication proxy feature so that users can have the configured authorization to permit traffic going through the configured interfaces.	no	no	no	no	no	yes

Table 37 Supported TACACS+ Authentication and Authorization AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1
route	Specifies a route to be applied to an interface. Used with service=slip, service=ppp, and protocol=ip. During network authorization, the route attribute can be used to specify a per-user static route, to be installed by TACACS+ as follows: `route="`*dst_address mask* `[`*gateway*`]"` This indicates a temporary static route that is to be applied. The *dst_address*, *mask*, and *gateway* are expected to be in the usual dotted-decimal notation, with the same meanings as in the familiar **ip route** configuration command on a network access server. If *gateway* is omitted, the peer's address is the gateway. The route is expunged when the connection terminates.	no	yes	yes	yes	yes	yes
route#<n>	Like the route AV pair, this specifies a route to be applied to an interface, but these routes are numbered, allowing multiple routes to be applied. Used with service=ppp and protocol=ip, and service=ppp and protocol=ipx.	no	no	no	yes	yes	yes
routing=x	Specifies whether routing information is to be propagated to and accepted from this interface. Used with service=slip, service=ppp, and protocol=ip. Equivalent in function to the /routing flag in SLIP and PPP commands. Can either be true or false (for example, routing=true).	yes	yes	yes	yes	yes	yes
rte-fltr-in#<n>	Specifies an input access list definition to be installed and applied to routing updates on the current interface for the duration of the current connection. Used with service=ppp and protocol=ip, and with service=ppp and protocol=ipx.	no	no	no	yes	yes	yes

Table 37 Supported TACACS+ Authentication and Authorization AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1
rte-fltr-out#<n>	Specifies an output access list definition to be installed and applied to routing updates on the current interface for the duration of the current connection. Used with service=ppp and protocol=ip, and with service=ppp and protocol=ipx.	no	no	no	yes	yes	yes
sap#<n>	Specifies static Service Advertising Protocol (SAP) entries to be installed for the duration of a connection. Used with service=ppp and protocol=ipx.	no	no	no	yes	yes	yes
sap-fltr-in#<n>	Specifies an input SAP filter access list definition to be installed and applied on the current interface for the duration of the current connection. Used with service=ppp and protocol=ipx.	no	no	no	yes	yes	yes
sap-fltr-out#<n>	Specifies an output SAP filter access list definition to be installed and applied on the current interface for the duration of the current connection. Used with service=ppp and protocol=ipx.	no	no	no	yes	yes	yes
send-auth	Defines the protocol to use (PAP or CHAP) for username-password authentication following CLID authentication.	no	no	no	no	no	yes
send-secret	Specifies the password that the NAS needs to respond to a chap/pap request from the remote end of a connection on an outgoing call.	no	no	no	no	no	yes
service=x	The primary service. Specifying a service attribute indicates that this is a request for authorization or accounting of that service. Current values are **slip**, **ppp**, **arap**, **shell**, **tty-daemon**, **connection**, and **system**. This attribute must always be included.	yes	yes	yes	yes	yes	yes

Table 37 Supported TACACS+ Authentication and Authorization AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1
source-ip=x	Used as the source IP address of all VPDN packets generated as part of a VPDN tunnel. This is equivalent to the Cisco **vpdn outgoing** global configuration command.	no	no	yes	yes	yes	yes
spi	Carries the authentication information needed by the home agent to authenticate a mobile node during registration. The information is in the same syntax as the **ip mobile secure host <addr>** configuration command. Basically it contains the rest of the configuration command that follows that string, verbatim. It provides the Security Parameter Index (SPI), key, authentication algorithm, authentication mode, and replay protection timestamp range.	no	no	no	no	no	yes
timeout=x	The number of minutes before an EXEC or ARA session disconnects (for example, timeout=60). A value of zero indicates no timeout. Used with service=arap.	yes	yes	yes	yes	yes	yes
tunnel-id	Specifies the username that will be used to authenticate the tunnel over which the individual user MID will be projected. This is analogous to the *remote name* in the **vpdn outgoing** command. Used with service=ppp and protocol=vpdn.	no	no	yes	yes	yes	yes
wins-servers=	Identifies a Windows NT server that can be requested by Microsoft PPP clients from the network access server during IPCP negotiation. To be used with service=ppp and protocol=ip. The IP address identifying each Windows NT server is entered in dotted decimal format.	no	no	no	yes	yes	yes
zonelist=x	A numeric zonelist value. Used with service=arap. Specifies an AppleTalk zonelist for ARA (for example, zonelist=5).	yes	yes	yes	yes	yes	yes

For more information about configuring TACACS+, refer to the "Configuring TACACS+" chapter. For more information about configuring TACACS+ authentication and authorization, refer to the "Configuring Authentication" and "Configuring Authorization" chapters.

TACACS+ Accounting AV Pairs

Table 38 lists and describes the supported TACACS+ accounting AV pairs, and it specifies the Cisco IOS release in which they are implemented.

Table 38 Supported TACACS+ Accounting AV Pairs

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1
Abort-Cause	If the fax session aborts, indicates the system component that signaled the abort. Examples of system components that could trigger an abort are FAP (Fax Application Process), TIFF (the TIFF reader or the TIFF writer), fax-mail client, fax-mail server, ESMTP client, or ESMTP server.	no	no	no	no	no	yes
bytes_in	The number of input bytes transferred during this connection.	yes	yes	yes	yes	yes	yes
bytes_out	The number of output bytes transferred during this connection.	yes	yes	yes	yes	yes	yes
Call-Type	Describes the type of fax activity: fax receive or fax send.	no	no	no	no	no	yes
cmd	The command the user executed.	yes	yes	yes	yes	yes	yes
data-rate	This AV pair has been renamed. See nas-rx-speed.						
disc-cause	Specifies the reason a connection was taken off-line. The Disconnect-Cause attribute is sent in accounting-stop records. This attribute also causes stop records to be generated without first generating start records if disconnection occurs before authentication is performed. Refer to Table 39 for a list of Disconnect-Cause values and their meanings.	no	no	no	yes	yes	yes
disc-cause-ext	Extends the disc-cause attribute to support vendor-specific reasons why a connection was taken off-line.	no	no	no	yes	yes	yes

Table 38 Supported TACACS+ Accounting AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1
elapsed_time	The elapsed time in seconds for the action. Useful when the device does not keep real time.	yes	yes	yes	yes	yes	yes
Email-Server-Address	Indicates the IP address of the e-mail server handling the on-ramp fax-mail message.	no	no	no	no	no	yes
Email-Server-Ack-Flag	Indicates that the on-ramp gateway has received a positive acknowledgment from the e-mail server accepting the fax-mail message.	no	no	no	no	no	yes
event	Information included in the accounting packet that describes a state change in the router. Events described are accounting starting and accounting stopping.	yes	yes	yes	yes	yes	yes
Fax-Account-Id-Origin	Indicates the account ID origin as defined by system administrator for the **mmoip aaa receive-id** or the **mmoip aaa send-id** command.	no	no	no	no	no	yes
Fax-Auth-Status	Indicates whether or not authentication for this fax session was successful. Possible values for this field are success, failed, bypassed, or unknown.	no	no	no	no	no	yes
Fax-Connect-Speed	Indicates the modem speed at which this fax-mail was initially transmitted or received. Possible values are 1200, 4800, 9600, and 14400.	no	no	no	no	no	yes
Fax-Coverpage-Flag	Indicates whether or not a cover page was generated by the off-ramp gateway for this fax session. True indicates that a cover page was generated; false means that a cover page was not generated.	no	no	no	no	no	yes
Fax-Dsn-Address	Indicates the address to which DSNs will be sent.	no	no	no	no	no	yes
Fax-Dsn-Flag	Indicates whether or not DSN has been enabled. True indicates that DSN has been enabled; false means that DSN has not been enabled.	no	no	no	no	no	yes
Fax-Mdn-Address	Indicates the address to which MDNs will be sent.	no	no	no	no	no	yes

Table 38 Supported TACACS+ Accounting AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1
Fax-Mdn-Flag	Indicates whether or not message delivery notification (MDN) has been enabled. True indicates that MDN had been enabled; false means that MDN had not been enabled.	no	no	no	no	no	yes
Fax-Modem-Time	Indicates the amount of time in seconds the modem sent fax data (x) and the amount of time in seconds of the total fax session (y), which includes both fax-mail and PSTN time, in the form x/y. For example, 10/15 means that the transfer time took 10 seconds, and the total fax session took 15 seconds.	no	no	no	no	no	yes
Fax-Msg-Id=	Indicates a unique fax message identification number assigned by Store and Forward Fax.	no	no	no	no	no	yes
Fax-Pages	Indicates the number of pages transmitted or received during this fax session. This page count includes cover pages.	no	no	no	no	no	yes
Fax-Process-Abort-Flag	Indicates that the fax session was aborted or successful. True means that the session was aborted; false means that the session was successful.	no	no	no	no	no	yes
Fax-Recipient-Count	Indicates the number of recipients for this fax transmission. Until e-mail servers support Session mode, the number should be 1.	no	no	no	no	no	yes
Gateway-Id	Indicates the name of the gateway that processed the fax session. The name appears in the following format: hostname.domain-name	no	no	no	no	no	yes
mlp-links-max	Gives the count of links which are known to have been in a given multilink session at the time the accounting record is generated.	no	no	no	yes	yes	yes
mlp-sess-id	Reports the identification number of the multilink bundle when the session closes. This attribute applies to sessions that are part of a multilink bundle. This attribute is sent in authentication-response packets.	no	no	no	yes	yes	yes

Table 38 Supported TACACS+ Accounting AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1
nas-rx-speed	Specifies the average number of bits per second over the course of the connection's lifetime. This attribute is sent in accounting-stop records.	no	no	no	yes	yes	yes
nas-tx-speed	Reports the transmit speed negotiated by the two modems.	no	no	no	yes	yes	yes
paks_in	The number of input packets transferred during this connection.	yes	yes	yes	yes	yes	yes
paks_out	The number of output packets transferred during this connection.	yes	yes	yes	yes	yes	yes
port	The port the user was logged in to.	yes	yes	yes	yes	yes	yes
Port-Used	Indicates the slot/port number of the Cisco AS5300 used to either transmit or receive this fax-mail.	no	no	no	no	no	yes
pre-bytes-in	Records the number of input bytes before authentication. This attribute is sent in accounting-stop records.	no	no	no	yes	yes	yes
pre-bytes-out	Records the number of output bytes before authentication. This attribute is sent in accounting-stop records.	no	no	no	yes	yes	yes
pre-paks-in	Records the number of input packets before authentication. This attribute is sent in accounting-stop records.	no	no	no	yes	yes	yes
pre-paks-out	Records the number of output packets before authentication. The Pre-Output-Packets attribute is sent in accounting-stop records.	no	no	no	yes	yes	yes
pre-session-time	Specifies the length of time, in seconds, from when a call first connects to when it completes authentication.	no	no	no	yes	yes	yes
priv_level	The privilege level associated with the action.	yes	yes	yes	yes	yes	yes
protocol	The protocol associated with the action.	yes	yes	yes	yes	yes	yes

Table 38 Supported TACACS+ Accounting AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1
reason	Information included in the accounting packet that describes the event that caused a system change. Events described are system reload, system shutdown, or when accounting is reconfigured (turned on or off).	yes	yes	yes	yes	yes	yes
service	The service the user used.	yes	yes	yes	yes	yes	yes
start_time	The time the action started (in seconds since the epoch, 12:00 a.m. Jan 1 1970). The clock must be configured to receive this information.	yes	yes	yes	yes	yes	yes
stop_time	The time the action stopped (in seconds since the epoch.) The clock must be configured to receive this information.	yes	yes	yes	yes	yes	yes
task_id	Start and stop records for the same event must have matching (unique) task_id numbers.	yes	yes	yes	yes	yes	yes
timezone	The time zone abbreviation for all timestamps included in this packet.	yes	yes	yes	yes	yes	yes
xmit-rate	This AV pair has been renamed. See nas-tx-speed.						

Table 39 lists the values and descriptions for the Disconnect Cause (disc-cause) attribute.

Table 39 Disconnect Cause Attribute Values

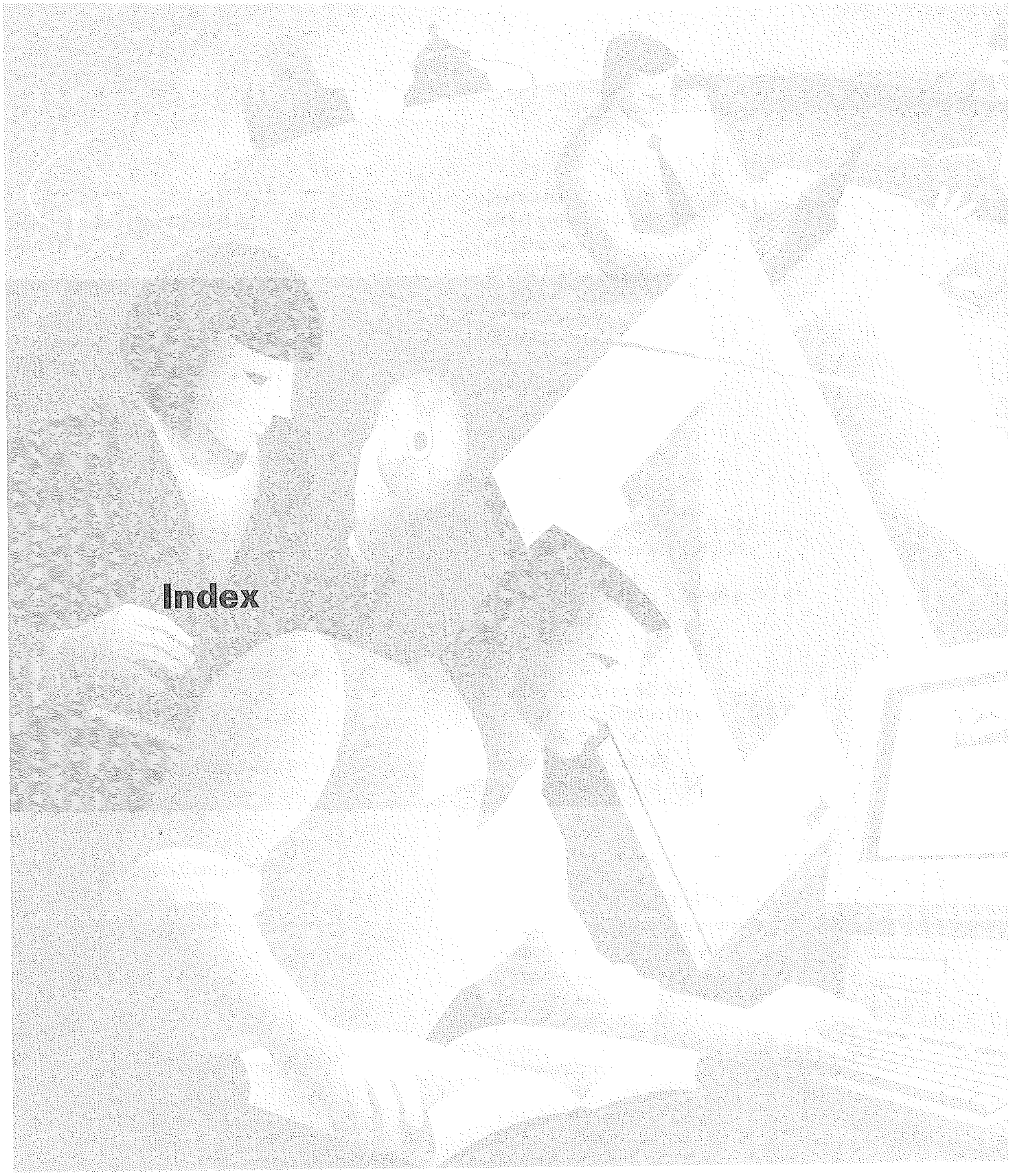
Value	Description
CLID-Authentication-Failure (4)	Failure to authenticate calling-party number.
Control-C-Detected (27)	Control-C detected. This value applies to EXEC sessions.
EXEC-Process-Destroyed (28)	EXEC process destroyed. This value applies to EXEC sessions.
Exit-Raw-TCP (24)	Disconnect due to exiting raw TCP. This value applies to EXEC sessions.
Exit-Telnet-Session (22)	Disconnect due to exiting Telnet session. This value applies to EXEC sessions.
Failed-PPP-CHAP-Auth (43)	PPP CHAP authentication failed. This value applies to PPP sessions.
Failed-PPP-LCP-Negotiation (41)	PPP LCP negotiation failed. This value applies to PPP sessions.
Failed-PPP-PAP-Auth-Fail (42)	PPP PAP authentication failed. This value applies to PPP sessions.
Failed-PPP-Remote-Auth (44)	PPP remote authentication failed. This value applies to PPP sessions.
Idle-Timeout (21)	Timeout waiting for user input. This value applies to all session types.
Invalid-Protocol (120)	Call refused because the detected protocol is disabled. This value applies to all session types.
Lost-Carrier (11)	Loss of carrier. This value applies to modem connections.
No-Carrier (10)	No carrier detected. This value applies to modem connections.

Table 39 Disconnect Cause Attribute Values (continued)

Value	Description
No-Detected-Result-Codes (12)	Failure to detect modem result codes. This value applies to modem connections.
No-Remote-IP-Addr (23)	Could not switch to SLIP/PPP; the remote end has no IP address. This value applies to EXEC sessions.
Password-Fail (25)	Bad passwords. This value applies to EXEC sessions.
PPP-Closed-Event (46)	Upper layer requested that the session be closed. This value applies to PPP sessions.
PPP-Remote-Terminate (45)	PPP received a Terminate Request from remote end. This value applies to PPP sessions.
Raw-TCP-Disabled (26)	Raw TCP disabled. This value applies to EXEC sessions.
Session-End-Callback (102)	Session terminated due to callback. This value applies to session types.
Session-Failed-Security (101)	Session failed for security reasons. This value applies to session types.
Session-Timeout (100)	Session timed out. This value applies to all session types.
Timeout-PPP-LCP (40)	PPP LCP negotiation timed out. This value applies to PPP sessions.
Unknown (2)	Reason unknown.
User-Ends-Session (20)	User terminates a session. This value applies to EXEC sessions.

For more information about configuring TACACS+, refer to the "Configuring TACACS+" chapter. For more information about configuring TACACS+ accounting, refer to the "Configuring Accounting" chapter.

Index



BC	Cisco IOS Bridging and IBM Networking Configuration Guide
DNC	Cisco IOS Dial Services Configuration Guide: Network Services
DTC	Cisco IOS Dial Services Configuration Guide: Terminal Services
FC	Cisco IOS Configuration Fundamentals Configuration Guide
IC	Cisco IOS Interface Configuration Guide
MC	Cisco IOS Multiservice Applications Configuration Guide
P1C	Cisco IOS IP and IP Routing Configuration Guide
P2C	Cisco IOS AppleTalk and Novell IPX Configuration Guide
P3C	Cisco IOS Apollo Domain, Banyan VINES, DECnet, ISO CLNS, and XNS Configuration Guide
QC	Cisco IOS Quality of Service Solutions Configuration Guide
SC	Cisco IOS Security Configuration Guide
WC	Cisco IOS Wide-Area Networking Configuration Guide
XC	Cisco IOS Switching Services Configuration Guide

CISCO SYSTEMS

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 69 18 61 00
Fax: 33 1 69 28 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/offices.

Argentina • Australia • Austria • Belgium • Brazil • Canada • Chile • China • Colombia • Costa Rica • Croatia • Czech Republic • Denmark • Dubai, UAE
Finland • France • Germany • Greece • Hong Kong • Hungary • India • Indonesia • Ireland • Israel • Italy • Japan • Korea • Luxembourg • Malaysia
Mexico • The Netherlands • New Zealand • Norway • Peru • Philippines • Poland • Portugal • Puerto Rico • Romania • Russia • Saudi Arabia • Singapore
Slovakia • Slovenia • South Africa • Spain • Sweden • Switzerland • Taiwan • Thailand • Turkey • Ukraine • United Kingdom • United States • Venezuela

Printed in the USA on recycled paper containing 10% postconsumer waste.

Cover Part Number: 56-2343-06