Cisco PIX Firewall Command Reference

Version 6.3

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: 78-14890-01
Text Part Number: 78-14890-01

CONTENTS

About This Guide

This preface introduces the *Cisco PIX Firewall Command Reference* and contains the following sections:

Document Objectives

This guide contians the commands available for use with the Cisco PIX Firewall to protect your network from unauthorized use and to establish Virtual Private Networks (VPNs) to connect remote sites and users to your network.

Audience

This guide is for network managers who perform any of the following tasks:

- Managing network security
- Configuring firewalls
- Managing default and static routes, and TCP and UDP services

Use this guide with the *Cisco PIX Firewall Hardware Installation Guide* and the *Cisco PIX Firewall and VPN Configuration Guide*.

Document Organization

This guide includes the following chapters:

- Chapter 1, "PIX Firewall Software Version 6.3 Commands," provides you with a quick reference to the commands available in the PIX Firewall software.
- Chapter 2, "Using PIX Firewall Commands," introduces you to the PIX Firewall commands, access modes, and common port and protocol numbers.
- Chapter 3, "A through B Commands," provides detailed descriptions of all commands that begin with the letters A or B.
- Chapter 4, "C Commands," provides detailed descriptions of all commands that begin with the letter C.
- Chapter 5, "D through F Commands," provides detailed descriptions of all commands that begin with the letters D through F.
- Chapter 6, "G through L Commands," provides detailed descriptions of all commands that begin with the letters G through L.
- Chapter 7, "M through R Commands," provides detailed descriptions of all commands that begin with the letters M through R.
- Chapter 8, "S Commands," provides detailed descriptions of all commands that begin with the letter S.
- Chapter 9, "T through Z Commands," provides detailed descriptions of all commands that begin with the letters T through X.

Document Conventions

The PIX Firewall command syntax descriptions use the following conventions:

Command descriptions use these conventions:

- Braces ({ }) indicate a required choice.
- Square brackets ([ ]) indicate optional elements.
- Vertical bars (|) separate alternative, mutually exclusive elements.
- **Boldface** indicates commands and keywords that are entered literally as shown.
- *Italics* indicate arguments for which you supply values.

Examples use these conventions:

- Examples depict screen displays and the command line in `screen` font.
- Information you need to enter in examples is shown in **`boldface screen`** font.
- Variables for which you must supply a value are shown in *`italic screen`* font.

Graphic user interface access uses these conventions:

- **Boldface** indicates buttons and menu items.
- Selecting a menu item (or screen) is indicated by the following convention:

  Click **Start>Settings>Control Panel**.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Related Documentation

Use this document in conjunction with the PIX Firewall documentation available online at the following site:

http://www.cisco.com/en/US/products/sw/secursw/ps2120/index.html

Obtaining Documentation

Cisco provides several ways to obtain documentation, technical assistance, and other technical resources. These sections explain how to obtain technical information from Cisco Systems.

Cisco.com

You can access the most current Cisco documentation on the World Wide Web at this URL:

http://www.cisco.com/univercd/home/home.htm

You can access the Cisco website at this URL:

http://www.cisco.com

International Cisco web sites can be accessed from this URL:

http://www.cisco.com/public/countries_languages.shtml

Documentation CD-ROM

Cisco documentation and additional literature are available in a Cisco Documentation CD-ROM package, which may have shipped with your product. The Documentation CD-ROM is updated monthly and may be more current than printed documentation. The CD-ROM package is available as a single unit or through an annual subscription.

Registered Cisco.com users can order the Documentation CD-ROM (product number DOC-CONDOCCD=) through the online Subscription Store:

http://www.cisco.com/go/subscription

Ordering Documentation

You can find instructions for ordering documentation at this URL:

http://www.cisco.com/univercd/cc/td/doc/es_inpck/pdi.htm

You can order Cisco documentation in these ways:

- Registered Cisco.com users (Cisco direct customers) can order Cisco product documentation from the Networking Products MarketPlace:

  http://www.cisco.com/en/US/partner/ordering/index.shtml

- Registered Cisco.com users can order the Documentation CD-ROM (Customer Order Number DOC-CONDOCCD=) through the online Subscription Store:

  http://www.cisco.com/go/subscription

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco Systems Corporate Headquarters (California, U.S.A.) at 408 526-7208 or, elsewhere in North America, by calling 800 553-NETS (6387).

Documentation Feedback

You can submit comments electronically on Cisco.com. On the Cisco Documentation home page, click **Feedback** at the top of the page.

You can e-mail your comments to bug-doc@cisco.com.

You can submit your comments by mail by using the response card behind the front cover of your document or by writing to the following address:

Cisco Systems
Attn: Customer Document Ordering
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Obtaining Technical Assistance

Cisco provides Cisco.com, which includes the Cisco Technical Assistance Center (TAC) Website, as a starting point for all technical assistance. Customers and partners can obtain online documentation, troubleshooting tips, and sample configurations from the Cisco TAC website. Cisco.com registered users have complete access to the technical support resources on the Cisco TAC website, including TAC tools and utilities.

Cisco.com

Cisco.com offers a suite of interactive, networked services that let you access Cisco information, networking solutions, services, programs, and resources at any time, from anywhere in the world.

Cisco.com provides a broad range of features and services to help you with these tasks:

- Streamline business processes and improve productivity
- Resolve technical issues with online support
- Download and test software packages
- Order Cisco learning materials and merchandise
- Register for online skill assessment, training, and certification programs

To obtain customized information and service, you can self-register on Cisco.com at this URL:

http://www.cisco.com

Technical Assistance Center

The Cisco TAC is available to all customers who need technical assistance with a Cisco product, technology, or solution. Two levels of support are available: the Cisco TAC website and the Cisco TAC Escalation Center. The avenue of support that you choose depends on the priority of the problem and the conditions stated in service contracts, when applicable.

We categorize Cisco TAC inquiries according to urgency:

- Priority level 4 (P4)—You need information or assistance concerning Cisco product capabilities, product installation, or basic product configuration.
- Priority level 3 (P3)—Your network performance is degraded. Network functionality is noticeably impaired, but most business operations continue.
- Priority level 2 (P2)—Your production network is severely degraded, affecting significant aspects of business operations. No workaround is available.
- Priority level 1 (P1)—Your production network is down, and a critical impact to business operations will occur if service is not restored quickly. No workaround is available.

Cisco TAC Website

You can use the Cisco TAC website to resolve P3 and P4 issues yourself, saving both cost and time. The site provides around-the-clock access to online tools, knowledge bases, and software. To access the Cisco TAC website, go to this URL:

http://www.cisco.com/tac

All customers, partners, and resellers who have a valid Cisco service contract have complete access to the technical support resources on the Cisco TAC website. Some services on the Cisco TAC website require a Cisco.com login ID and password. If you have a valid service contract but do not have a login ID or password, go to this URL to register:

http://tools.cisco.com/RPF/register/register.do

If you are a Cisco.com registered user, and you cannot resolve your technical issues by using the Cisco TAC website, you can open a case online at this URL:

http://www.cisco.com/en/US/support/index.html

If you have Internet access, we recommend that you open P3 and P4 cases through the Cisco TAC website so that you can describe the situation in your own words and attach any necessary files.

Cisco TAC Escalation Center

The Cisco TAC Escalation Center addresses priority level 1 or priority level 2 issues. These classifications are assigned when severe network degradation significantly impacts business operations. When you contact the TAC Escalation Center with a P1 or P2 problem, a Cisco TAC engineer automatically opens a case.

To obtain a directory of toll-free Cisco TAC telephone numbers for your country, go to this URL:

http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml

Before calling, please check with your network operations center to determine the level of Cisco support services to which your company is entitled: for example, SMARTnet, SMARTnet Onsite, or Network Supported Accounts (NSA). When you call the center, please have available your service agreement number and your product serial number.

Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

- *The Cisco Product Catalog* describes the networking products offered by Cisco Systems as well as ordering and customer support services. Access the *Cisco Product Catalog* at this URL:

  http://www.cisco.com/en/US/products/products_catalog_links_launch.html

- Cisco Press publishes a wide range of networking publications. Cisco suggests these titles for new and experienced users: *Internetworking Terms and Acronyms Dictionary, Internetworking Technology Handbook, Internetworking Troubleshooting Guide, and the Internetworking Design Guide.* For current Cisco Press titles and other information, go to Cisco Press online at this URL:

  http://www.ciscopress.com

- *Packet* magazine is the Cisco monthly periodical that provides industry professionals with the latest information about the field of networking. You can access *Packet* magazine at this URL:

  http://www.cisco.com/en/US/about/ac123/ac114/about_cisco_packet_magazine.html

- *iQ Magazine* is the Cisco monthly periodical that provides business leaders and decision makers with the latest information about the networking industry. You can access *iQ Magazine* at this URL:

  http://business.cisco.com/prod/tree.taf%3fasset_id=44699&public_view=true&kbns=1.html

- *Internet Protocol Journal* is a quarterly journal published by Cisco Systems for engineering professionals involved in the design, development, and operation of public and private internets and intranets. You can access the *Internet Protocol Journal* at this URL:

  http://www.cisco.com/en/US/about/ac123/ac147/about_cisco_the_internet_protocol_journal.html

- Training—Cisco offers world-class networking training, with current offerings in network training listed at this URL:

  http://www.cisco.com/en/US/learning/le31/learning_recommended_training_list.html

CHAPTER 1

PIX Firewall Software Version 6.3 Commands

Table 1-1 lists the commands that are supported in PIX Firewall software Version 6.3.

Table 1-1 Supported Commands

A-D	E-M	M-S	S (continued)-Z
aaa accounting	eeprom	mtu	show history
aaa authentication	enable	multicast	show local-host/clear local host
aaa authorization	established	name/names	show memory
aaa-server	exit	nameif	show processes
access-group	failover	nat	show tech-support
access-list	filter	ntp	show traffic/clear traffic
activation-key	fixup protocol	object-group	show uauth/clear uauth
alias	fixup protocol snmp	outbound/apply	show version
arp	floodguard	pager	show xlate/clear xlate
auth-prompt	fragment	password	shun
auto-update	global	pdm	· When this feature is off, regular SIP Fixup will work as it does under PIX 6.3.3
banner	help	perfmon	ssh
ca	hostname	ping	static
ca generate rsa key	http	prefix-list	sysopt
capture	icmp	privilege	telnet
clear	igmp	quit	terminal
clock	interface	reload	tftp-server
conduit	ip address	rip	timeout
configure	ip audit	route	url-block
console	ip local pool	route-map	url-cache
copy	ip verify reverse-path	router ospf	url-server
crypto dynamic-map	isakmp	routing interface	username

Table 1-1 Supported Commands (continued)

A-D	E-M	M-S	S (continued)-Z
crypto ipsec	isakmp policy	service	virtual
crypto map	kill	session enable	vpdn
debug	logging	setup	vpnclient
dhcpd	login	show	vpngroup
dhcprelay	mac-list	show blocks/clear blocks	who
disable	management-access	show checksum	write
domain-name	mgcp	show conn	
dynamic-map	mroute	show cpu usage	

CHAPTER 2

Using PIX Firewall Commands

This chapter introduces the *Cisco PIX Firewall Command Reference* and contains the following sections:

Introduction

This section provides a brief introduction to using PIX Firewall commands and where to go for more information on configuring and using your PIX Firewall.

The following table lists some basic PIX Firewall commands.

Task	Related Command
Saving my configuration	**write memory**
Viewing my configuration	**write terminal**
Accumulating system log (syslog) messages	**logging buffered debugging**
Viewing system log (syslog) messages	**show logging**
Clearing the message buffer	**clear logging**

Tips

Tip When using the PIX Firewall command-line interface (CLI), you can do the following:

- Check the syntax before entering a command. Enter a command and press the **Enter** key to view a quick summary, or precede a command with **help**, as in, **help aaa**.
- Abbreviate commands. For example, you can use the **config t** command to start configuration mode, the **write t** command statement to list the configuration, and the **write m** command to write to Flash memory. Also, in most commands, **show** can be abbreviated as **sh**. This feature is called command completion.
- After changing or removing the **alias**, **access-list**, **conduit**, **global, nat**, **outbound**, and **static** commands, use the **clear xlate** command to make the IP addresses available for access.
- Review possible port and protocol numbers at the following IANA websites:

  http://www.iana.org/assignments/port-numbers
  http://www.iana.org/assignments/protocol-numbers
- Create your configuration in a text editor and then cut and paste it into the configuration. PIX Firewall lets you paste in a line at a time or the whole configuration. Always check your configuration after pasting large blocks of text to be sure everything copied.

For more information

For information about how to build your PIX Firewall configuration, please refer to the *Cisco PIX Firewall and VPN Configuration Guide*.

Syslog messages are fully described in *Cisco PIX Firewall System Log Messages*.

For information about how to use Cisco PIX Device Manager (PDM), please refer to the online Help included in the PDM software (accessed through the PDM application Help button). For information about how to install PDM, please refer to the *Cisco PIX Device Manager Installation Guide*.

PIX Firewall technical documentation is located online at the following website:

http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/

Command Modes

The PIX Firewall contains a command set based on Cisco IOS technologies and provides configurable command privilege modes based on the following command modes:

- Unprivileged mode. When you first access the firewall, it displays the ">" prompt. This is unprivileged mode, and it lets you view firewall settings. The unprivileged mode prompt appears as follows:

```
pixfirewall>
```

- Privileged mode, which displays the "#" prompt and lets you change current settings. Any unprivileged mode command also works in privileged mode. Use the **enable** command to start privileged mode from unprivileged mode as follows:

```
pixfirewall> enable
Password:
pixfirewall#
```

  Use the **exit** or **quit** commands to exit privileged mode and return to unprivileged mode as follows:

```
pixfirewall# exit

Logoff

Type help or '?' for a list of available commands.
pixfirewall>
```

  Use the **disable** command to exit privileged mode and return to unprivileged mode as follows:

```
pixfirewall# disable
pixfirewall>
```

- Configuration mode, which displays the "(config)#" prompt and lets you change the firewall configuration. All privileged, unprivileged, and configuration mode commands are available in this mode. Use the **configure terminal** command to start configuration mode as follows:

```
pixfirewall# configure terminal
pixfirewall(config)#
```

  Use the **exit** or **quit** commands to exit configuration mode and return to privileged mode as follows:

```
pixfirewall(config)# quit
pixfirewall#
```

  Use the **disable** command to exit configuration mode and return to unprivileged mode as follows:

```
pixfirewall(config)# disable
pixfirewall>
```

Ports

Literal names can be used instead of a numerical port value in **access-list** commands.

The PIX Firewall uses port 1521 for SQL*Net. This is the default port used by Oracle for SQL*Net; however, this value does not agree with IANA port assignments.

The PIX Firewall listens for RADIUS on ports 1645 and 1646. If your RADIUS server uses ports 1812 and 1813, you must reconfigure it to listen on ports 1645 and 1646.

To assign a port for DNS access, use **domain**, not **dns**. The **dns** keyword translates into the port value for **dnsix**.

Note By design, the PIX Firewall drops DNS packets sent to UDP port 53 (usually used for DNS) that have a packet size larger than 512 bytes.

Port numbers can be viewed online at the IANA website:

http://www.iana.org/assignments/port-numbers

Table 2-1 lists the port literal values.

Table 2-1 Port Literal Values

Literal	TCP or UDP?	Value	Description
aol	TCP	5190	America On-line
bgp	TCP	179	Border Gateway Protocol, RFC 1163
biff	UDP	512	Used by mail system to notify users that new mail is received
bootpc	UDP	68	Bootstrap Protocol Client
bootps	UDP	67	Bootstrap Protocol Server
chargen	TCP	19	Character Generator
citrix-ica	TCP	1494	Citrix Independent Computing Architecture (ICA) protocol
cmd	TCP	514	Similar to **exec** except that **cmd** has automatic authentication
ctiqbe	TCP	2748	Computer Telephony Interface Quick Buffer Encoding
daytime	TCP	13	Day time, RFC 867
discard	TCP, UDP	9	Discard
domain	TCP, UDP	53	DNS (Domain Name System)
dnsix	UDP	195	DNSIX Session Management Module Audit Redirector
echo	TCP, UDP	7	Echo
exec	TCP	512	Remote process execution
finger	TCP	79	Finger
ftp	TCP	21	File Transfer Protocol (control port)
ftp-data	TCP	20	File Transfer Protocol (data port)
gopher	TCP	70	Gopher
https	TCP	443	Hyper Text Transfer Protocol (SSL)
h323	TCP	1720	H.323 call signalling
hostname	TCP	101	NIC Host Name Server
ident	TCP	113	Ident authentication service
imap4	TCP	143	Internet Message Access Protocol, version 4
irc	TCP	194	Internet Relay Chat protocol

Table 2-1 Port Literal Values (continued)

Literal	TCP or UDP?	Value	Description
isakmp	UDP	500	Internet Security Association and Key Management Protocol
kerberos	TCP, UDP	750	Kerberos
klogin	TCP	543	KLOGIN
kshell	TCP	544	Korn Shell
ldap	TCP	389	Lightweight Directory Access Protocol
ldaps	TCP	636	Lightweight Directory Access Protocol (SSL)
lpd	TCP	515	Line Printer Daemon - printer spooler
login	TCP	513	Remote login
lotusnotes	TCP	1352	IBM Lotus Notes
mobile-ip	UDP	434	MobileIP-Agent
nameserver	UDP	42	Host Name Server
netbios-ns	UDP	137	NetBIOS Name Service
netbios-dgm	UDP	138	NetBIOS Datagram Service
netbios-ssn	TCP	139	NetBIOS Session Service
nntp	TCP	119	Network News Transfer Protocol
ntp	UDP	123	Network Time Protocol
pcanywhere-status	UDP	5632	pcAnywhere status
pcanywhere-data	TCP	5631	pcAnywhere data
pim-auto-rp	TCP, UDP	496	Protocol Independent Multicast, reverse path flooding, dense mode
pop2	TCP	109	Post Office Protocol - Version 2
pop3	TCP	110	Post Office Protocol - Version 3
pptp	TCP	1723	Point-to-Point Tunneling Protocol
radius	UDP	1645	Remote Authentication Dial-In User Service
radius-acct	UDP	1646	Remote Authentication Dial-In User Service (accounting)
rip	UDP	520	Routing Information Protocol
secureid-udp	UDP	5510	SecureID over UDP
smtp	TCP	25	Simple Mail Transport Protocol
snmp	UDP	161	Simple Network Management Protocol
snmptrap	UDP	162	Simple Network Management Protocol - Trap
sqlnet	TCP	1521	Structured Query Language Network
ssh	TCP	22	Secure Shell
sunrpc (rpc)	TCP, UDP	111	Sun Remote Procedure Call
syslog	UDP	514	System Log

Table 2-1 Port Literal Values (continued)

Literal	TCP or UDP?	Value	Description
tacacs	TCP, UDP	49	Terminal Access Controller Access Control System Plus
talk	TCP, UDP	517	Talk
telnet	TCP	23	RFC 854 Telnet
tftp	UDP	69	Trivial File Transfer Protocol
time	UDP	37	Time
uucp	TCP	540	UNIX-to-UNIX Copy Program
who	UDP	513	Who
whois	TCP	43	Who Is
www	TCP	80	World Wide Web
xdmcp	UDP	177	X Display Manager Control Protocol

Protocols

Literal names can be used instead of a numerical port value in **access-list** commands.

Protocol numbers can be viewed online at the IANA website:

http://www.iana.org/assignments/port-numbers

Note Many routing protocols use multicast packets to transmit their data. If you send routing protocols across the PIX Firewall, configure the surrounding routers with the Cisco IOS software **neighbor** command. If routes on an unprotected interface are corrupted, the routes transmitted to the protected side of the firewall will pollute routers there as well.

The PIX Firewall supports the protocol literal values listed in Table 2-2 .

Table 2-2 Protocol Literal Values

Literal	Value	Description
ah	51	Authentication Header for IPv6, RFC 1826
eigrp	88	Enhanced Interior Gateway Routing Protocol
esp	50	Encapsulating Security Payload (ESP) for IPv6, RFC 1827
gre	47	General routing encapsulation
icmp	1	Internet Control Message Protocol, RFC 792
igmp	2	Internet Group Management Protocol, RFC 1112
igrp	9	Interior Gateway Routing Protocol
ipinip	4	IP-in-IP encapsulation
nos	94	Network Operating System (Novell NetWare)
ospf	89	Open Shortest Path First routing protocol, RFC 1247

Table 2-2 Protocol Literal Values (continued)

Literal	Value	Description
pcp	108	Payload Compression Protocol
snp	109	Sitara Networks Protocol
tcp	6	Transmission Control Protocol, RFC 793
udp	17	User Datagram Protocol, RFC 768

Deprecated Commands

The following commands are no longer used to configure the firewall: **sysopt route dnat**, **sysopt security fragguard**, **fragguard,** and **session enable**.

The **sysopt route dnat** command is ignored, starting in PIX Firewall software Version 6.2. Instead, overlapping configurations (network addresses and routes) are automatically handled by outside NAT.

The **sysopt security fragguard** and **fragguard** commands have been replaced by the **fragment** command.

The **session enable** command is deprecated because the AccessPro router it was intended to support no longer exists.

CHAPTER 3

A through B Commands

aaa accounting

Enable, disable, or view LOCAL, TACACS+, or RADIUS user accounting (on a server designated by the **aaa-server** command).

[**no**] **aaa accounting include | exclude** *service if_name local_ip local_mask foreign_ip foreign_mask server_tag*

[**no**] **aaa accounting include | exclude** *service if_name server_tag*

clear aaa [**accounting include | exclude** *service if_name server_tag*]

[**no**] **aaa accounting match** *acl_name if_name server_tag*

show aaa

Syntax Description

accounting	Enable or disable accounting services. Use of this command requires that you previously used the **aaa-server** command to designate a AAA server.
exclude	Create an exception to a previously stated rule by excluding the specified service from accounting. The **exclude** parameter improves the former **except** option by allowing the user to specify a port to exclude to a specific host or hosts.
foreign_ip	The IP address of the hosts you want to access the *local_ip* address. Use 0 to mean all hosts.
foreign_mask	Network mask of *foreign_ip*. Always specify a specific mask value. Use 0 if the IP address is 0. Use 255.255.255.255 for a host.
if_name	Interface name from which users require authentication. Use *if_name* in combination with the *local_ip* address and the *foreign_ip* address to determine where access is sought and from whom. The *local_ip* address is always on the highest security level interface and *foreign_ip* is always on the lowest.
include	Create a new rule with the specified service to include.
local_ip	The IP address of the host or network of hosts that you want to be authenticated or authorized. You can set this address to **0** to mean all hosts and to let the authentication server decide which hosts are authenticated.
local_mask	Network mask of *local_ip*. Always specify a specific mask value. Use 0 if the IP address is 0. Use 255.255.255.255 for a host.

match *acl_name*	Specify an **access-list** command statement name.
server_tag	The AAA server group tag defined by the **aaa-server** command. To use the local PIX Firewall user authentication database, enter **LOCAL** for this parameter.
service	The accounting service. Accounting is provided for all services or you can limit it to one or more services. Possible values are **any**, **ftp**, **http**, **telnet**, or *protocol*/*port*. Use **any** to provide accounting for all TCP services. To provide accounting for UDP services, use the *protocol*/*port* form. For *protocol*/*port*, the TCP *protocol* appears as 6, the UDP protocol appears as 17, and so on, and port is the TCP or UDP destination port. A port value of 0 (zero) means all ports. For protocols other than TCP and UDP, the *port* is not applicable and should not be used.

Defaults

For *protocol*/*port*, the TCP *protocol* appears as 6, the UDP protocol appears as 17, and so on, and port is the TCP or UDP destination port. A port value of 0 (zero) means all ports. For protocols other than TCP and UDP, the *port* is not applicable and should not be used.

Command Modes

Configuration mode.

Usage Guidelines

User accounting services keep a record of which network services a user has accessed. These records are also kept on the designated AAA server. Accounting information is only sent to the active server in a server group.

Use the **aaa accounting** command with the **aaa authentication** and **aaa authorization** commands.

The **include** and **exclude** options are not backward compatible with previous PIX Firewall versions. If you downgrade to an earlier version, the **aaa** command statements will be removed from your configuration.

Note Traffic that is not specified by an **include** statement is not processed.

For outbound connections, first use the **nat** command to determine which IP addresses can access the PIX Firewall. For inbound connections, first use the **static** and **access-list** command statements to determine which inside IP addresses can be accessed through the PIX Firewall from the outside network.

If you want to allow connections to come from any host, code the local IP address and netmask as **0.0.0.0 0.0.0.0**, or **0 0**. The same convention applies to the foreign host IP address and netmask; **0.0.0.0 0.0.0.0** means any foreign host.

Tip The **help aaa** command displays the syntax and usage for the **aaa authentication**, **aaa authorization**, **aaa accounting**, and **aaa proxy-limit** commands in summary form.

Examples

The default PIX Firewall configuration provides the following **aaa-server** protocols:

```
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
aaa-server LOCAL protocol local
```

The following example uses the default protocol TACACS+ with the **aaa** commands:

```
aaa-server TACACS+ (inside) host 10.1.1.10 thekey timeout 20
aaa authentication include any outbound 0 0 0 0 TACACS+
aaa authorization include any outbound 0 0 0 0
aaa accounting include any outbound 0 0 0 0 TACACS+
aaa authentication serial console TACACS+
```

This example specifies that the authentication server with the IP address 10.1.1.10 resides on the inside interface and is in the default TACACS+ server group. The next three command statements specify that any users starting outbound connections to any foreign host will be authenticated using TACACS+, that the users who are successfully authenticated are authorized to use any service, and that all outbound connection information will be logged in the accounting database. The last command statement specifies that access to the PIX Firewall unit's serial console requires authentication from the TACACS+ server.

Related Commands

aaa authentication	Enables, disables, or displays LOCAL, TACACS+, or RADIUS user authentication on a server designated by the **aaa-server** command, or for PDM user authentication.
aaa authorization	Enables or disables LOCAL or TACACS+ user authorization services.
auth-prompt	Changes the AAA challenge text.
password	Sets the password for Telnet access to the PIX Firewall console.
service	Resets inbound connections.
ssh	Specifies a host for access through Secure Shell (SSH).
telnet	Specifies the host for access via Telnet.
virtual	Accesses the PIX Firewall virtual server.

aaa authentication

Enable, disable, or view LOCAL, TACACS+, or RADIUS user authentication, on a server designated by the **aaa-server** command, or PDM user authentication.

[**no**] **aaa authentication include** | **exclude** *authen_service if_name local_ip local_mask* [*foreign_ip foreign_mask*] *server_tag*

clear aaa [**authentication include** | **exclude** *authen_service if_name local_ip local_mask foreign_ip foreign_mask server_tag*]

[**no**] **aaa authentication match** *acl_name if_name server_tag*

[**no**] **aaa authentication secure-http-client**

[**no**] **aaa authentication** [**serial** | **enable** | **telnet** | **ssh** | **http**] **console** *server_tag* [**LOCAL**]

show aaa

Syntax Description

authen_service	**Specifies the type of traffic to include or exclude from authentication based on the service option selected.** **access authentication** The access authentication service options are as follows: **enable**, **serial**, **ssh**, and **telnet**. Specify **serial** for serial console access, **telnet** for Telnet access, **ssh** for SSH access, and **enable** for enable-mode access. **cut-through authentication** The cut-through authentication service options are as follows: **telnet**, **ftp**, **http**, **https**, **icmp/***type*, *proto*, **tcp/***port*, and **udp/***port*. The variable *proto* can be any supported IP protocol value or name: for example, **ip** or **igmp**. Only Telnet, FTP, HTTP, or HTTPS traffic triggers interactive user authentication. You can enter an ICMP message type number for *type* to include or exclude that specific ICMP message type from authentication. For example, **icmp/8** includes or excludes type 8 (echo request) ICMP messages. The **tcp/0** option enables authentication for all TCP traffic, which includes FTP, HTTP, HTTPS, and Telnet. When a specific *port* is specified, only the traffic with a matching destination port is included or excluded for authentication. Note that FTP, Telnet, HTTP, and HTTPS are equivalent to **tcp/21**, **tcp/23**, **tcp/80**, and **tcp/443**, respectively. If **ip** is specified, all IP traffic is included or excluded for authentication, depending on whether **include** or **exclude** is specified. When all IP traffic is included for authentication, following are the expected behaviors: • Before a user (source IP-based) is authenticated, an FTP, Telnet, HTTP, or HTTPS request triggers authentication and all other IP requests are denied. • After a user is authenticated through FTP, Telnet, HTTP, HTTPS, or virtual Telnet authentication (see the **virtual** command), all traffic is free from authentication until the **uauth** timeout.
authentication	Enable or disable user authentication, prompt user for username and password, and verify information with authentication server. When used with the **console** option, enables or disables authentication service for access to the PIX Firewall console over Telnet or from the Console connector on the PIX Firewall unit. Use of the **aaa authentication** command requires that you previously used the **aaa-server** command to designate an authentication server. The **aaa authentication** command supports HTTP authentication. The PIX Firewall requires authentication verification of the HTTP server through the **aaa authentication http console** command before PDM can access the PIX Firewall.
console	Specify that access to the PIX Firewall console require authentication and optionally, log configuration changes to a syslog server. The maximum password length for accessing the console is 16 characters.
enable	Access verification for the PIX Firewall unit's privilege mode.

exclude	Create an exception to a previously stated rule by excluding the specified service from authentication. The **exclude** parameter improves the former **except** option by allowing the user to specify a port to exclude to a specific host or hosts.
foreign_ip	The IP address of the hosts you want to access the *local_ip* address. Use 0 to mean all hosts.
foreign_mask	Network mask of *foreign_ip*. Always specify a specific mask value. Use 0 if the IP address is 0. Use 255.255.255.255 for a host.
http	Access verification for the HTTP (Hypertext Transfer Protocol) access to the PIX Firewall (via PDM). The maximum username prompt for HTTP authentication is 30 characters. The maximum password length is 15 characters.
if_name	The interface name from which to authenticate users.
include	Create a new rule with the specified service to include.
local_ip	The IP address of the host or network of hosts that you want to be authenticated or authorized. You can set this address to **0** to mean all hosts and to let the authentication server decide which hosts are authenticated.
local_mask	Network mask of *local_ip*. Always specify a specific mask value. Use 0 if the IP address is 0. Use 255.255.255.255 for a host.
match *acl_name*	Specify an **access-list** command statement name. However, do not use an **access-list** command statement that uses the source port to identify matching traffic. Like the **aaa authentication include | exclude** command, the source port is not supported in the match criteria of the **aaa authentication match** *acl_name* command.
secure-http-client	Secures HTTP client authentication (through SSL) for HTTP cut-through proxy authentication.
serial	Access verification for the PIX Firewall unit's serial console.

server_tag	The AAA server group tag defined by the **aaa-server** command.
	
	For cut-through proxy and "to the box" authentication, you can also use the local PIX Firewall user authentication database by specifying the server group tag **LOCAL**. If **LOCAL** is specified for *server_tag* and the local user credential database is empty, the following warning message appears:
	
	`Warning:local database is empty! Use 'username' command to define local users.`
	
	Conversely, if the local database becomes empty when **LOCAL** is still present in the command, the following warning message appears:
	
	`Warning:Local user database is empty and there are still commands using 'LOCAL' for authentication.`

ssh	Access verification for the SSH access to the PIX Firewall console.
telnet	Access verification for the Telnet access to the PIX Firewall console.

Defaults

If a **aaa authentication http console** *server_tag* command statement is not defined, you can gain access to the PIX Firewall (via PDM) with no username and the PIX Firewall enable password (set with the **password** command). If the **aaa** commands are defined but the HTTP authentication requests a time out, which implies the AAA servers may be down or not available, you can gain access to the PIX Firewall using the username **pix** and the enable password. By default, the enable password is not set.

The PIX Firewall supports authentication usernames up to 127 characters and passwords of up to 16 characters (some AAA servers accept passwords up to 32 characters). A password or username may not contain an "@" character as part of the password or username string, with a few exceptions.

Tip The **help aaa** command displays the syntax and usage for the **aaa authentication**, **aaa authorization**, **aaa accounting**, and **aaa proxy-limit** commands in summary form.

The authentication ports supported for AAA are fixed. We support port 21 for FTP, port 23 for Telnet, and port 80 for HTTP. For this reason, do not use Static PAT to reassign ports for services you wish to authenticate. In other words, when the port to authenticate is not one of the three known ports, the firewall rejects the connection instead of authenticating it.

Command Modes

Configuration mode.

Usage Guidelines

To use the **aaa authentication** command, you must first designate an authentication server with the **aaa-server** command. Also, for each IP address, one **aaa authentication** command is permitted for inbound connections and one for outbound connections.

Use the *if_name*, *local_ip*, and *foreign_ip* variables to define where access is sought and from whom. The address for *local_ip* is always on the highest security level interface and *foreign_ip* is always on the lowest.

The **aaa authentication** command is not intended to mandate your security policy. The authentication servers determine whether a user can or cannot access the system, what services can be accessed, and what IP addresses the user can access. The PIX Firewall interacts with FTP, HTTP , HTTPS, and Telnet to display the credentials prompts for logging in to the network or logging in to exit the network. You can specify that only a single service be authenticated, but this must agree with the authentication server to ensure that both the firewall and server agree.

The **include** and **exclude** options are not backward compatible with previous PIX Firewall versions. If you downgrade to an earlier version, these **aaa authentication** command statements will be removed from your configuration.

Note When a cut-through proxy is configured, TCP sessions (TELNET, FTP, HTTP, or HTTPS) may have their sequence number randomized even if the **norandomseq** option is used in the **nat** or **static** command. This occurs when a AAA server proxies the TCP session to authenticate the user before permitting access.

aaa authentication console command

The **aaa authentication serial console** command enables you to require authentication verification to access the PIX Firewall unit's serial console. The **serial console** options also logs to a syslog server changes made to the configuration from the serial console.

Authenticated access to the PIX Firewall console has different types of prompts depending on the option you choose with the **aaa authentication** [**serial | enable | telnet | ssh**] **console** server_tag **[LOCAL]** command. While the **enable** and **ssh** options allow three tries before stopping with an access denied message, both the **serial** and **telnet** options cause the user to be prompted continually until successfully logging in. The **serial** option requests a username and password before the first command line prompt on the serial console connection. The **telnet** option forces you to specify a username and password before the first command line prompt of a Telnet console connection. The **enable** option requests a

username and password before accessing privileged mode for serial, Telnet, or SSH connections. The **ssh** option requests a username and password before the first command line prompt on the SSH console connection. The **ssh** option allows a maximum of three authentication attempts. The [**LOCAL**] keyword option specifies a second authentication method that can be local only.

Telnet access to the PIX Firewall console is available from any internal interface, and from the outside interface with IPSec configured, and requires previous use of the **telnet** command. SSH access to the PIX Firewall console is also available from any interface without IPSec configured, and requires previous use of the **ssh** command.

The new **ssh** option specifies the group of AAA servers to be used for SSH user authentication. The authentication protocol and AAA server IP addresses are defined with the **aaa-server** command statement.

Similar to the Telnet model, if a **aaa authentication ssh console** *server_tag* command statement is not defined, you can gain access to the PIX Firewall console with the username **pix** and with the PIX Firewall Telnet password (set with the **passwd** command). If the **aaa** command is defined but the SSH authentication requests timeouts, which implies the AAA servers may be down or not available, you can gain access to the PIX Firewall using username **pix** and the enable password (set with the **enable password** command). By default, the Telnet password is **cisco** and the enable password is not set.

If the console login request times out, you can gain access to the PIX Firewall from the serial console by entering the username **pix** and the enable password.

The **LOCAL** keyword is optional when specified as a RADIUS or TACACS+ server only. Any access to the module (**SSH, Telnet, enable**) requiring a username and password is prompted only three times.

If an **aaa authentication ssh console** *server_tag* command is not defined, you can gain access to the CLI with the username **pix** and with the PIX Telnet password (set with the **passwd** command). If the **aaa** command is defined but the SSH authentication requests timeouts, which implies that the AAA servers may be down or not available, you can gain access to the PIX Firewall using the username **pix** and the enable password (set with the **enable password** command).

The PIX Firewall supports authentication usernames up to 127 characters and passwords up to 16 characters (some AAA servers accept passwords up to 32 characters). A password or username may not contain an "@" character as part of the password or username string.

The command only accepts the second, optional **LOCAL** keyword when the *server_tag* refers to an existing, valid TACACS+ or RADIUS server group defined in a **aaa-server** command. You can configure **LOCAL** as the first and only *server_tag*.

The **no** form of the command removes the complete command and does not support removing single methods.

aaa authentication secure-http-client

The **aaa authentication secure-http-client** command enables SSL and secures username and password exchange between HTTP clients and the firewall. It offers a secure method for user authentication to the firewall prior to allowing the user's HTTP-based web requests to traverse the firewall.

The following example configures HTTP traffic to be authenticated securely:

```
aaa authentication secure-http-client
aaa authentication include http ...
```

where "..." represents your values for *authen_service if_name local_ip local_mask* [*foreign_ip foreign_mask*] *server_tag*.

The following are limitations of the **aaa authentication secure-http-client** command:

- At runtime, a maximum of 16 HTTPS authentication processes are allowed. If all 16 HTTPS authentication processes are running, the 17th, new HTTPS connection requiring authentication is dropped.
- When **uauth timeout 0** is configured (the **uauth timeout** is set to 0), HTTPS authentication may not work. If a browser initiates multiple TCP connections to load a web page after HTTPS authentication, the first connection is let through but the subsequent connections trigger authentication. As a result, users are presented with an authentication page, continuously, even if the correct username and password are entered each time. You can workaround this by setting the **uauth timeout** to 1 second with the **timeout uauth 0:0:1** command. However, this workaround opens a 1-second window of opportunity that may allow non-authenticated users to go through the firewall if they are comming from the same source IP address.
- Because HTTPS authentication occurs on the SSL port 443, users must not configure an **access-list** command statement to block traffic from the HTTP client to HTTP server on port 443. Furthermore, if static PAT is configured for web traffic on port 80, it must also be configured for the SSL port. In the following example, the first line configures static PAT for web traffic and the second line must be added to support the HTTPS authentication configuration:

```
static (inside,outside) tcp 10.132.16.200 www 10.130.16.10 www
static (inside,outside) tcp 10.132.16.200 443 10.130.16.10 443
```

Enabling Authentication

The **aaa authentication** command enables or disables the following features:

- User authentication services provided by a TACACS+ or RADIUS server are first designated with the **aaa authorization** command. A user starting a connection via FTP, Telnet, or over the World Wide Web is prompted for their username and password. If the username and password are verified by the designated TACACS+ or RADIUS authentication server, the PIX Firewall unit will allow further traffic between the authentication server and the connection to interact independently through the PIX Firewall unit's "cut-through proxy" feature.
- Administrative authentication services providing access to the PIX Firewall unit's console via Telnet, SSH, or the serial console. Telnet access requires previous use of the **telnet** command. SSH access requires previous use of the **ssh** command.

The prompts users see requesting AAA credentials differ between the three services that can access the PIX Firewall for authentication: Telnet, FTP, HTTP, and HTTPS:

- Telnet users see a prompt generated by the PIX Firewall that you can change with the **auth-prompt** command. The PIX Firewall permits a user up to four chances to log in and then if the username or password still fails, the PIX Firewall drops the connection.
- FTP users receive a prompt from the FTP program. If a user enters an incorrect password, the connection is dropped immediately. If the username or password on the authentication database differs from the username or password on the remote host to which you are using FTP to access, enter the username and password in these formats:

  authentication_user_name@*remote_system_user_name*
  authentication_password@*remote_system_password*

  If you daisy-chain PIX Firewall units, Telnet authentication works in the same way as a single unit, but FTP and HTTP authentication have additional complexity for users because they have to enter each password and username with an additional at (@) character and password or username for each daisy-chained system. Users can exceed the 63-character password limit depending on how many units are daisy-chained and password length.

  Some FTP graphical user interfaces (GUIs) do not display challenge values.

- HTTP users see a pop-up window generated by the browser itself if **aaa authentication secure-http-client** is not configured. If **aaa authentication secure-http-client** is configured, a form will load in the browser which is designed to collect username and password. In either case, if a user enters an incorrect password, the user is reprompted. When the web server and the authentication server are on different hosts, use the **virtual** command to get the correct authentication behavior.

Authenticated access to the PIX Firewall console has different types of prompts depending on the option you choose with the **aaa authentication console** command:

- **enable** option—Allows three tries before stopping with "Access denied." The **enable** option requests a username and password before accessing privileged mode for serial or Telnet connections.
- **serial** option—Causes the user to be prompted continually until successfully logging in. The **serial** option requests a username and password before the first command line prompt on the serial console connection.
- **ssh** option—Allows three tries before stopping with "Rejected by Server." The **ssh** option requests a username and password before the first command line prompt appears.
- **telnet** option—Causes the user to be prompted continually until successfully logging in. The **telnet** option forces you to specify a username and password before the first command line prompt of a Telnet console connection.

You can specify an interface name with the **aaa authentication** command. In previous versions, if you specified **aaa authentication include any outbound 0 0** *server*, PIX Firewall only authenticated outbound connections and not those to the perimeter interface. PIX Firewall now authenticates any outbound connection to the outside as well as to hosts on the perimeter interface. To preserve the behavior of previous versions, use these commands to enable authentication and to disable authentication from the inside to the perimeter interface:

```
aaa authentication include any outbound 0 0 server
aaa authentication exclude outbound perim_net perim_mask server
```

When a host is configured for authentication, all users on the host must use a web browser or Telnet first before performing any other networking activity, such as accessing mail or a news reader. The reason for this is that users must first establish their authentication credentials and programs such as mail agents and newsreaders do not have authentication challenge prompts.

The PIX Firewall only accepts 7-bit characters during authentication. After authentication, the client and server can negotiate for 8 bits if required. During authentication, the PIX Firewall only negotiates Go-Ahead, Echo, and NVT (network virtual terminal).

HTTP Authentication

When using HTTP authentication to a site running Microsoft IIS that has "Basic text authentication" or "NT Challenge" enabled, users may be denied access from the Microsoft IIS server. This occurs because the browser appends the string: "Authorization: Basic=Uuhjksdkfhk==" to the HTTP GET commands. This string contains the PIX Firewall authentication credentials.

Windows NT Microsoft IIS servers respond to the credentials and assume that a Windows NT user is trying to access privileged pages on the server. Unless the PIX Firewall username password combination is exactly the same as a valid Windows NT username and password combination on the Microsoft IIS server, the HTTP GET command is denied.

To solve this problem, PIX Firewall provides the **virtual http** command, which redirects the browser's initial connection to another IP address, authenticates the user, then redirects the browser back to the URL which the user originally requested.

Once authenticated, a user never has to reauthenticate no matter how low the PIX Firewall uauth timeout is set. This is because the browser caches the "Authorization: Basic=Uuhjksdkfhk==" string in every subsequent connection to that particular site. This can *only* be cleared when the user exits *all* instances of Netscape Navigator or Internet Explorer and restarts. Flushing the cache is of no use.

As long as the user repeatedly browses the Internet, the browser resends the "Authorization: Basic=Uuhjksdkfhk==" string to transparently reauthenticate the user.

Multimedia applications such as CU-SeeMe, Intel Internet Phone, MeetingPoint, and MS NetMeeting silently start the HTTP service before an H.323 session is established from the inside to the outside.

Network browsers such as Netscape Navigator do not present a challenge value during authentication; therefore, only password authentication can be used from a network browser.

Note To avoid interfering with these applications, do not enter blanket outgoing **aaa** command statements for all challenged ports such as using the **any** option. Be selective with which ports and addresses you use to challenge HTTP, and when to set user authentication timeouts to a higher timeout value. If interfered with, the multimedia programs may fail on the PC and may even crash the PC after establishing outgoing sessions from the inside.

TACACS+ and RADIUS servers

Up to 196 TACACS+ or RADIUS servers are permitted (up to 14 servers in each of the up to 14 server groups—set with the **aaa-server** command). When a user logs in, the servers are accessed one at a time starting with the first server you specify in the configuration, until a server responds.

The PIX Firewall permits only one authentication type per network. For example, if one network connects through the PIX Firewall using TACACS+ for authentication, another network connecting through the PIX Firewall can authenticate with RADIUS, but one network cannot authenticate with both TACACS+ and RADIUS.

For the TACACS+ server, if you do not specify a key to the **aaa-server** command, no encryption occurs.

The PIX Firewall displays the same timeout message for both RADIUS and TACACS+. The message "aaa server host machine not responding" displays when either of the following occurs:

- The AAA server system is down.
- The AAA server system is up, but the service is not running.

Previously, TACACS+ differentiated between the two preceding states and provided two different timeout messages, while RADIUS did not differentiate between the two states and provided one timeout message.

aaa authentication match

The **aaa authentication match** *acl_name interface_name server_tag* command specifies to match an **access-list** command statement and then to provide authentication for that match. However, do not use an **access-list** command statement that uses the source port to identify matching traffic. Like the **aaa authentication include | exclude** command, the source port is not supported in the match criteria of the **aaa authentication match** *acl_name* command.

The following set of examples illustrates how to use this command, as follows:

```
show access-list
access-list mylist permit tcp 10.0.0.0 255.255.255.0 172.23.2.0 255.255.255.0
access-list yourlist permit tcp any any
show aaa
aaa authentication match mylist outbound TACACS+
```

Similar to IPSec, the keyword **permit** means "yes" and **deny** means "no." Therefore, the following command,

```
aaa authentication match yourlist outbound tacacs
```

is equal to this command:

```
aaa authentication include any outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 tacacs
```

The **aaa** command statement list is order-dependent between **access-list** command statements. If the following command is entered:

```
aaa authentication match yourlist outbound tacacs
```

after this command:

```
aaa authentication match mylist outbound TACACS+
```

The PIX Firewall tries to find a match in the **mylist access-list** command statement group before it tries to find a match in the **yourlist access-list** command statement group.

Old **aaa** command configuration and functionality stays the same and is not converted to the **access-list** command format. Hybrid access control configurations (that is, old configurations combined with new **access-list** command-based configurations) are not recommended.

Examples

The following example shows use of the **aaa authentication** command:

```
pixfirewall(config) aaa authentication telnet console radius
```

The following example lists the new include and exclude options:

```
aaa authentication include any outbound 172.31.0.0 255.255.0.0 0.0.0.0 0.0.0.0 tacacs+
aaa authentication exclude telnet outbound 172.31.38.0 255.255.255.0 0.0.0.0 0.0.0.0
tacacs+
```

The following examples demonstrate ways to use the *if_name* parameter. The PIX Firewall has an inside network of 192.168.1.0, an outside network of 209.165.201.0 (subnet mask 255.255.255.224), and a perimeter network of 209.165.202.128 (subnet mask 255.255.255.224).

This example enables authentication for connections originated from the inside network to the outside network:

```
aaa authentication include any outbound 192.168.1.0 255.255.255.0 209.165.201.0
255.255.255.224 tacacs+
```

This example enables authentication for connections originated from the inside network to the perimeter network:

```
aaa authentication include any outbound 192.168.1.0 255.255.255.0 209.165.202.128
255.255.255.224 tacacs+
```

This example enables authentication for connections originated from the outside network to the inside network:

```
aaa authentication include any inbound 192.168.1.0 255.255.255.0 209.165.201.0
255.255.255.224 tacacs+
```

This example enables authentication for connections originated from the outside network to the perimeter network:

```
aaa authentication include any inbound 209.165.201.0 255.255.255.224 209.165.202.128
255.255.255.224 tacacs+
```

This example enables authentication for connections originated from the perimeter network to the outside network:

```
aaa authentication include any outbound 209.165.202.128 255.255.255.224 209.165.201.0
255.255.255.224 tacacs+
```

This example specifies that IP addresses 10.0.0.1 through 10.0.0.254 can originate outbound connections and then enables user authentication so that those addresses must enter user credentials to exit the PIX Firewall. In this example, the first **aaa authentication** command permits authentication on FTP, HTTP, or Telnet depending on what the authentication server handles. The second **aaa authentication** command lets host 10.0.0.42 start outbound connections without being authenticated. This example uses the default authentication group **tacacs+**.

```
nat (inside) 1 10.0.0.0 255.255.255.0
aaa authentication include any outbound 0 0 tacacs+
aaa authentication exclude outbound 10.0.0.42 255.255.255.255 tacacs+ any
```

This example permits inbound access to any IP address in the range of 209.165.201.1 through 209.165.201.30 indicated by the 209.165.201.0 network address (subnet mask 255.255.255.224). All services are permitted by the **access-list** command, and the **aaa authentication** command permits authentication on FTP, HTTP, or Telnet depending on what the authentication server handles. The authentication server is at IP address 10.16.1.20 on the inside interface.

```
aaa-server AuthIn protocol tacacs+
aaa-server AuthIn (inside) host 10.16.1.20 thisisakey timeout 20
static (inside,outside) 209.165.201.0 10.16.1.0 netmask 255.255.255.224
access-list acl_out permit tcp 10.16.1.0 255.255.255.0 209.165.201.0 255.255.255.224
access-group acl_out in interface outside
aaa authentication include any inbound 0 0 AuthIn
```

Related Commands

aaa authorization	Enable or disable LOCAL or TACACS+ user authorization services.
auth-prompt	Changes the AAA challenge text.
password	Sets the password for Telnet access to the PIX Firewall console.
service	Resets inbound connections.
ssh	Specifies a host for access through Secure Shell (SSH).
telnet	Specifies the host for access via Telnet.
virtual	Accesses the PIX Firewall virtual server.

aaa authorization

Enable or disable LOCAL or TACACS+ user authorization services.

[**no**] **aaa authorization command {LOCAL |** *tacacs_server_tag***}**

[**no**] **aaa authorization include | exclude** *svc if_name local_ip local_mask foreign_ip foreign_mask*

clear aaa [**authorization** [**include | exclude** *svc if_name local_ip local_mask foreign_ip foreign_mask*]]

[**no**] **aaa authorization match** *acl_name if_name server_tag*

show aaa

Syntax Description

authorization	Enable or disable TACACS+ user authorization for services (PIX Firewall does not support RADIUS authorization). The authentication server determines what services the user is authorized to access.
exclude	Create an exception to a previously stated rule by excluding the specified service from authentication, authorization, or accounting to the specified host. The **exclude** parameter improves the former **except** option by allowing the user to specify a port to exclude to a specific host or hosts.
foreign_ip	The IP address of the hosts you want to access the *local_ip* address. Use 0 to mean all hosts.
foreign_mask	Network mask of *foreign_ip*. Always specify a specific mask value. Use 0 if the IP address is 0. Use 255.255.255.255 for a host.
if_name	Interface name from which users require authentication. Use *if_name* in combination with the *local_ip* address and the *foreign_ip* address to determine where access is sought and from whom. The *local_ip* address is always on the highest security level interface and *foreign_ip* is always on the lowest.
include	Create a new rule with the specified service to include.
LOCAL	Specifies use of the PIX Firewall local user database for local command authorization (using privilege levels). The command will only accept the second, optional LOCAL method when the <*server_tag*> refers to an existing, valid AAA TACACS+ or RADIUS server group defined in a **aaa-server** configuration command. Clearly, you can configure LOCAL as the first and only <*server_tag*>. The no form of the command will remove the complete command and will not support removing single methods.
local_ip	The IP address of the host or network of hosts that you want to be authenticated or authorized. You can set this address to **0** to mean all hosts and to let the authentication server decide which hosts are authenticated.
local_mask	Network mask of *local_ip*. Always specify a specific mask value. Use 0 if the IP address is 0. Use 255.255.255.255 for a host.
match *acl_name*	Specify an **access-list** command statement name.
server_tag	The AAA server group tag as defined by the **aaa-server** command. You can also enter **LOCAL** for the group tag value and use the local firewall database AAA services such as local command authorization privilege levels.

svc	The services which require authorization. Use **any**, **ftp**, **http**, **telnet**, or *protocol/port*. Use **any** to provide authorization for all TCP services. To provide authorization for UDP services, use the *protocol/port* form. Services not specified are authorized implicitly. (Services specified in the **aaa authentication** command do not affect the services that require authorization.) For *protocol/port*: • *protocol*—the protocol (6 for TCP, 17 for UDP, 1 for ICMP, and so on). • *port*—the TCP or UDP destination port, or port range. The *port* can also be the ICMP type; that is, 8 for ICMP echo or ping. A port value of 0 (zero) means all ports. Port ranges only applies to the TCP and UDP protocols, not to ICMP. For protocols other than TCP, UDP, and ICMP the *port* is not applicable and should not be used. An example port specification follows. `aaa authorization include udp/53-1024 inside 0 0 0 0` This example enables authorization for DNS lookups to the inside interface for all clients, and authorizes access to any other services that have ports in the range of 53 to 1024. **Note** Specifying a port range may produce unexpected results at the authorization server. PIX Firewall sends the port range to the server as a string with the expectation that the server will parse it out into specific ports. Not all servers do this. In addition, you may want users to be authorized on specific services, which will not occur if a range is accepted.
tacacs_server_tag	Specifies to use a TACACS user authentication server.

Defaults

An IP address of **0** means all hosts.

Command Modes

Configuration mode.

Usage Guidelines

Except for its use with command authorization, the **aaa authorization** command requires previous configuration with the **aaa authentication** command; however, use of the **aaa authentication** command does not require use of a **aaa authorization** command.

Currently, the **aaa authorization** command is supported for use with LOCAL and TACACS+ servers but not with RADIUS servers.

Tip The **help aaa** command displays the syntax and usage for the **aaa authentication**, **aaa authorization**, **aaa accounting**, and **aaa proxy-limit** commands in summary form.

For each IP address, one **aaa authorization** command is permitted. If you want to authorize more than one service with **aaa authorization**, use the **any** parameter for the service type.

If the first attempt at authorization fails and a second attempt causes a timeout, use the **service resetinbound** command to reset the client that failed the authorization so that it will not retransmit any connections. An example authorization timeout message in Telnet follows.

```
Unable to connect to remote host: Connection timed out
```

User authorization services control which network services a user can access. After a user is authenticated, attempts to access restricted services cause the PIX Firewall unit to verify the access permissions of the user with the designated AAA server.

The **include** and **exclude** options are not backward compatible with previous PIX Firewall versions. If you downgrade to an earlier version, the **aaa** command statements will be removed from your configuration.

Note RADIUS authorization is supported for use with **access-list** command statements and for use in configuring a RADIUS server with an **acl=***acl_name* vendor-specific identifier. Refer to the **access-list** command page for more information. Also see the **aaa-server radius-authport** commands.

If the AAA console login request times out, you can gain access to the PIX Firewall from the serial console by entering the **pix** username and the enable password.

Examples

The default PIX Firewall configuration provides the following **aaa-server** protocols:

```
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
```
aaa-server LOCAL protocol local

The following example uses the default protocol TACACS+ with the **aaa** commands:

```
aaa-server TACACS+ (inside) host 10.1.1.10 thekey timeout 20
aaa authentication include any outbound 0 0 0 0 TACACS+
aaa authorization include any outbound 0 0 0 0
aaa accounting include any outbound 0 0 0 0 TACACS+
aaa authentication serial console TACACS+
```

This example specifies that the authentication server with the IP address 10.1.1.10 resides on the inside interface and is in the default TACACS+ server group. The next three command statements specify that any users starting outbound connections to any foreign host will be authenticated using TACACS+, that the users who are successfully authenticated are authorized to use any service, and that all outbound connection information will be logged in the accounting database. The last command statement specifies that access to the PIX Firewall unit's serial console requires authentication from the TACACS+ server.

The following example enables authorization for DNS lookups from the outside interface:

```
aaa authorization include udp/53 inbound 0.0.0.0 0.0.0.0
```

The following example enables authorization of ICMP echo-reply packets arriving at the inside interface from inside hosts:

```
aaa authorization include 1/0 outbound 0.0.0.0 0.0.0.0
```

This means that users will not be able to ping external hosts if they have not been authenticated using Telnet, HTTP, or FTP.

The following example enables authorization for ICMP echoes (pings) only that arrive at the inside interface from an inside host:

```
aaa authorization include 1/8 outbound 0.0.0.0 0.0.0.0
```

Related Commands		
	aaa authentication	Enables, disables, or displays LOCAL, TACACS+, or RADIUS user authentication on a server designated by the **aaa-server** command, or for PDM user authentication.
	auth-prompt	Changes the AAA challenge text.
	password	Sets the password for Telnet access to the PIX Firewall console.
	service	Resets inbound connections.
	ssh	Specifies a host for access through Secure Shell (SSH).
	telnet	Specifies the host for access via Telnet.
	virtual	Accesses the PIX Firewall virtual server.

aaa mac-exempt

Exempts a list of MAC addresses from authentication and authorization.

[**no**] **aaa mac-exempt match** *id*

Syntax Description		
	id	A MAC access list number. (Configured with the **mac-list** command.)

Defaults

None.

Command Modes

The **aaa mac-exempt match** *id* command is available in configuration mode.

Usage Guidelines

The **aaa mac-exempt match** *id* command exempts a list of MAC addresses from authentication and authorization.

Note When configuring **mac-exempt**, do not use the same IP address for two MACs. If a **mac-exempt** command is configured for two MACs, M1 and M2, and both attempt to use the same ip address, only the traffic from M1 would be permitted. If a **mac-exempt** is configured for M1 or M2, or if one of them is not configured at all, then the traffic from second host would be allowed to pass. A syslog alerting you to a possible spoof attack, is generated.

Examples

The following example shows how to configure MAC-based AAA:

```
pixfirewall(config)# show mac-list
mac-list adc permit 00a0.c95d.0282 ffff.ffff.ffff
mac-list adc deny 00a1.c95d.0282 ffff.ffff.ffff
mac-list ac permit 0050.54ff.0000 ffff.ffff.0000
mac-list ac deny 0061.54ff.b440 ffff.ffff.ffff
mac-list ac deny 0072.54ff.b440 ffff.ffff.ffff

pixfirewall(config)# aaa mac-exempt match ac

pixfirewall(config)# show aaa
```

```
aaa mac-exempt match ac

pixfirewall(config)# aaa ?
Usage: [no] aaa authentication|authorization|accounting include|exclude <svc>
       <if_name><l_ip> <l_mask> [<f_ip> <f_mask>] <server_tag>
       [no] aaa authentication serial|telnet|ssh|http|enable console <server_tag>
       [no] aaa authentication|authorization|accounting match <acl_name> <if_name>
       <server_tag>
       [no] aaa authorization command {LOCAL | tacacs_server_tag} aaa proxy-limit <proxy
       limit> | disable
       [no] aaa mac-exempt match <mcl-id>
```

Related Commands

aaa authentication	Enable, disable, or view LOCAL, TACACS+, or RADIUS user authentication, on a server designated by the **aaa-server** command, or PDM user authentication.
aaa authorization	Enable or disable LOCAL or TACACS+ user authorization services.
access-list	Create an access list, or use downloadable access lists. (Downloadable access lists are supported for RADIUS servers only.)
mac-list	Adds a list of MAC addresses using a first match search, and used by the firewall VPN client in performing MAC-based authentication.

aaa proxy-limit

Specifies the number of concurrent proxy connections allowed per user.

[**no**] **aaa proxy-limit** *proxy_limit* | **disable**

show aaa proxy-limit

Syntax Description

disable	Disables the proxy limit.
proxy_limit	Specifies the number of concurrent proxy connections allowed per user, from 1 to 128. (The default value is 16.)

Defaults

The default proxy limit value is 16.

Command Modes

Configuration mode.

Usage Guidelines

The **aaa proxy-limit** command enables you to manually configure the uauth session limit by setting the maximum number of concurrent proxy connections allowed per user. By default, this value is set to 16. If a source address is a proxy server, consider excluding this IP address from authentication or increasing the number of allowable outstanding AAA requests.

The **show aaa proxy-limit** command displays the number of outstanding authentication requests allowed, or indicates that the proxy limit is disabled if disabled.

Examples

The following example shows how to set and display the maximum number of outstanding authentication requests allowed:

```
pixfirewall(config)# aaa proxy-limit 6
pixfirewall(config)# show aaa proxy-limit
aaa proxy-limit 6
```

Related Commands

aaa authentication	Enable, disable, or view LOCAL, TACACS+, or RADIUS user authentication, on a server designated by the **aaa-server** command, or PDM user authentication
aaa authorization	Enable or disable LOCAL or TACACS+ user authorization services.
aaa-server	Specifies a AAA server.

aaa-server

Defines the AAA server group.

[no] aaa-server server_tag deadtime <*minutes*>

[**no**] **aaa-server** *server_tag* [(*if_name*)] **host** *server_ip* [*key*] [**timeout** *seconds*]

[no] aaa-server *server_tag* **max-failed-attempts** <*number*>

[**no**] **aaa-server** *server_tag* **protocol** *auth_protocol*

[**no**] **aaa-server radius-acctport** [*acct_port*]

[**no**] **aaa-server radius-authport** [*auth_port*]

clear aaa-server [*server_tag*]

show aaa-server

debug radius session

Syntax Description

aaa-server	Specifies a AAA server or up to 14 groups of servers with a maximum of 14 servers each. Certain types of AAA services can be directed to different servers. Services can also be set up to fail over to multiple servers.
*acct*_port	RADIUS authentication port number. The default is 1645.
*auth*_port	RADIUS accounting port number. The default is 1646.
deadtime <*minutes*>	<*minutes*> identifies the minutes to declare the AAA server group as unresponsive.
debug radius session	Captures RADIUS session information and attributes for sent and received RADIUS packets.

host *server_ip*	The IP address of the TACACS+ or RADIUS server.
if_name	The interface name on which the server resides.
key	A case-sensitive, alphanumeric keyword of up to 127 characters that is the same value as the key on the TACACS+ server. Any characters entered past 127 are ignored. The key is used between the client and server for encrypting data between them. The *key* must be the same on both the client and server systems. Spaces are not permitted in the key, but other special characters are.
max-failed-attempts <*number*>	<*number*> identifies the maximum number of AAA requests to attempt to each AAA server in a AAA server group.
no aaa-server	Unbinds a AAA server from and interface or host.
protocol *auth_protocol*	The type of AAA server, either **tacacs+** or **radius**.
radius-acctport	Sets the port number of the RADIUS server which the PIX Firewall unit will use for accounting functions. The default port number used for RADIUS accounting is **1646**.
radius-authport	Sets the port number of the RADIUS server which the PIX Firewall will use for authentication functions. The default port number used for RADIUS authentication is **1645**.
server_tag	An alphanumeric string which is the name of the server group. Use the *server_tag* in the **aaa** command to associate **aaa authentication** and **aaa accounting** command statements to a AAA server. Up to 14 server groups are permitted. However, **LOCAL** cannot used with **aaa-server** command because **LOCAL** is predefined by the PIX Firewall.
timeout *seconds*	The timeout interval for the request. This is the time after which the PIX Firewall gives up on the request to the primary AAA server. If there is a standby AAA server, the PIX Firewall will send the request to the backup server. The retransmit timeout is currently set to 10 seconds and is not user configurable.

Defaults

By default, the PIX Firewall listens for RADIUS on ports **1645** for authentication and **1646** for accounting. (The default ports 1645 for authentication and 1646 for accounting are as defined in RFC 2058.)

The default configuration provides the following **aaa-server** command protocols:

```
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
```
aaa-server LOCAL protocol local

The default timeout value is 5 seconds.

Some AAA servers accept passwords up to 32 characters, but the PIX Firewall allows passwords up to 16 characters only.

Command Modes

Configuration mode.

Usage Guidelines

The **aaa-server** command lets you specify AAA server groups. PIX Firewall lets you define separate groups of TACACS+ or RADIUS servers for specifying different types of traffic; such as, a TACACS+ server for inbound traffic and another for outbound traffic. Another use is where all outbound HTTP traffic will be authenticated by a TACACS+ server, and all inbound traffic will use RADIUS.

Other **aaa** commands reference the server tag group defined by the **aaa-server** command *server_tag* parameter. This is a global setting that takes effect when the TACACS+ or RADIUS service is started.

Note When a cut-through proxy is configured, TCP sessions (TELNET, FTP, or HTTP) may have their sequence number randomized even if the **norandomseq** option is used in the **nat** or **static** command. This occurs when a AAA server proxies the TCP session to authenticate the user before permitting access.

AAA server groups are defined by a tag name that directs different types of traffic to each authentication server. If the first authentication server in the list fails, the AAA subsystem fails over to the next server in the tag group. You can have up to 14 tag groups and each group can have up to 14 AAA servers for a total of up to 196 AAA servers.

If accounting is in effect, the accounting information goes only to the active server.

The **show aaa-server** command displays AAA server configuration.

[no] aaa-server server_tag **deadtime** <minutes>

The *server_tag* identifies the AAA server group and is the same as the current **aaa-server** command.

<*minutes*> identifies the minutes to declare the AAA server group as unresponsive

Valid input rage: 0 - 1440
Units: minutes
Default:10

While the command may be configured even without having configured the LOCAL method on any of the three authentication and authorization commands described earlier, it only affects operations when a user has configured two methods. Obviously, at this time, the second method must and be *LOCAL*.

The command specifies the minutes a particular method should be marked unresponsive and skipped. When a AAA server group has been marked unresponsive, the firewall will immediately perform the authentication or authorization against the next method which will be the local firewall user database. Every server in a group must be marked unresponsive before the entire group will be declared unresponsive.

When you configure the deadtime to “0”, the AAA server group is never considered unresponsive and all authentication and authorization requests are always attempted against this AAA server group first before using the next method in the method list (for example, falling back to the local user database).

The **[no]** form of this command restores the **aaa-server** command to its default value of 10 minutes.

The *deadtime* begins as soon as the last server in the AAA server group has been marked DOWN. A server is marked down when maximum number of attempts defined in max-attempts has been reached and failed to receive a response. Upon expiration of the deadtime, the AAA server group becomes active and all requests will are submitted once again to the AAA servers in the AAA server group.

[no] aaa-server server_tag **max-failed-attempts** <number>

The *server_tag* identifies the AAA server group and is the same as existing **aaa-server** command today.

<*number*> identifies the maximum number of AAA requests to attempt to each AAA server in a AAA server group.

Valid input rage: 1 -5
Units: Counter
Default: 3 (same as current PIX/FWSM software)

The current PIX/FWSM software sends a AAA request 3 times to a AAA server before it declares the AAA server unresponsive and moves on to try the next server in the group. This command lets the user configure this number of attempts. Users should tune the *max-failed-attempts* and the timeout values to achieve the desired fall-back behavior when authenticating or authorizing commands in a fall-back configuration. That is, if you wish to declare an individual AAA server unresponsive more aggressively, you should reduce the *max-failed-attempts* counter to 1 or 2.

aaa-server radius-authport and aaa-server radius-acctport

You can change authorization and accounting port settings on the firewall with the **aaa-server radius-authport** and **aaa-server radius-acctport** commands. These commands specify the destination TCP/UDP port number of the remote RADIUS server host to which you wish to assign authentication or accounting functions.

By default, the PIX Firewall listens for RADIUS on ports 1645 and 1646. If your authentication server uses ports other than 1645 and 1646, then you must configure the firewall for the appropriate ports prior to starting the RADIUS service with the **aaa-server** command. For example, some RADIUS servers use the port numbers 1812 and 1813 as defined in RFC 2138 and RFC 2139. If your RADIUS server uses ports 1812 and 1813, you must use the **aaa-server radius-authport** and **aaa-server radius-acctport** commands to reconfigure the firewall to use ports 1812 and 1813.

The following port pairs are listed as assigned to authentication and accounting services on RADIUS servers:

- 1645 (authentication), 1646 (accounting) - default for PIX Firewall
- 1812 (authentication), 1813 (accounting) - alternate

You can view these and other commonly used port number assignments online at the following website:

http://www.iana.org/assignments/port-numbers

Or, alternately, refer to "Ports" in Chapter 2, "Using PIX Firewall Commands," for additional information.

Upgrading Your AAA Server Configuration and Backward Compatibility

If you are upgrading from a previous version of PIX Firewall and have **aaa** command statements in your configuration, using the default server groups lets you maintain backward compatibility with the **aaa** command statements in your configuration.

The previous server type option at the end of the **aaa authentication** and **aaa accounting** commands has been replaced with the **aaa-server** *server_tag* group tag. Backward compatibility with previous versions is maintained by the inclusion of two default protocols for TACACS+ and RADIUS.

Examples

The following example uses the default protocol TACACS+ with the **aaa** commands:

```
aaa-server TACACS+ (inside) host 10.1.1.10 thekey timeout 20
aaa authentication include any outbound 0 0 0 0 TACACS+
aaa authorization include any outbound 0 0 0 0
aaa accounting include any outbound 0 0 0 0 TACACS+
aaa authentication serial console TACACS+
```

This example specifies that the authentication server with the IP address 10.1.1.10 resides on the inside interface and is in the default TACACS+ server group. The next three command statements specify that any users starting outbound connections to any foreign host will be authenticated using TACACS+, that

the users who are successfully authenticated are authorized to use any service, and that all outbound connection information will be logged in the accounting database. The last command statement specifies that access to the PIX Firewall unit's serial console requires authentication from the TACACS+ server.

This example creates the AuthOut and AuthIn server groups for RADIUS authentication and specifies that servers 10.0.1.40, 10.0.1.41, and 10.1.1.2 on the inside interface provide authentication. The servers in the AuthIn group authenticate inbound connections, the AuthOut group authenticates outbound connections.

```
aaa-server AuthIn protocol radius
aaa-server AuthIn (inside) host 10.0.1.40 ab timeout 20
aaa-server AuthIn (inside) host 10.0.1.41 abc timeout 4
aaa-server AuthOut protocol radius
aaa-server AuthOut (inside) host 10.1.1.2 abc123 timeout 15
aaa authentication include any inbound 0 0 0 0 AuthIn
aaa authentication include any outbound 0 0 0 0 AuthOut
```

The following example lists the commands that can be used to establish an Xauth crypto map:

```
ip address inside 10.0.0.1 255.255.255.0
ip address outside 168.20.1.5 255.255.255.0
ip local pool dealer 10.1.2.1-10.1.2.254
nat (inside) 0 access-list 80
aaa-server TACACS+ host 10.0.0.2 secret123
crypto ipsec transform-set pc esp-des esp-md5-hmac
crypto dynamic-map cisco 4 set transform-set pc
crypto map partner-map 20 ipsec-isakmp dynamic cisco
crypto map partner-map client configuration address initiate
crypto map partner-map client authentication TACACS+
crypto map partner-map interface outside
isakmp key cisco1234 address 0.0.0.0 netmask 0.0.0.0
isakmp client configuration address-pool local dealer outside
isakmp policy 8 authentication pre-share
isakmp policy 8 encryption des
isakmp policy 8 hash md5
isakmp policy 8 group 1
isakmp policy 8 lifetime 86400
```

The **aaa-server** command is used with the **crypto map** command to establish an authentication association so that VPN clients are authenticated when they access the PIX Firewall.

Related Commands

aaa authentication	Enable, disable, or view LOCAL, TACACS+, or RADIUS user authentication, on a server designated by the **aaa-server** command, or PDM user authentication.
aaa authorization	Enable or disable LOCAL or TACACS+ user authorization services.
crypto ipsec	Creates, displays, or deletes IPSec security associations, security association global lifetime values, and global transform sets.
isakmp	Negotiates IPSec security associations and enables IPSec secure communications.

access-group

Binds the access list to an interface.

[**no**] **access-group** *access-list* **in interface** *interface_name* [*per-user-override*]

clear access-group [*access-list*]

show access-group [*access-list*]

Syntax Description

access-list	The access list *id*.
in interface	Filter inbound packets at the given interface.
interface_name	The name of the network interface.
[per-user-override]	Allow downloadable user access lists to override the access list applied to the interface.

Defaults

None.

Command Modes

Configuration mode.

Usage Guidelines

The access-group command binds an access list to an interface. The access list is applied to traffic inbound to an interface. If you enter the **permit** option in an **access-list** command statement, the PIX Firewall continues to process the packet. If you enter the **deny** option in an **access-list** command statement, PIX Firewall discards the packet and generates the following syslog message.

```
%PIX-4-106019: IP packet from source_addr to destination_addr, protocol protocol received
from interface interface_name deny by access-group id
```

PIX Firewall Version 6.3(2) adds support for the **per-user-override** option, which allows downloaded access lists to override the access list applied to the interface. If the **per-user-override** optional argument is not present, thePIX Firewall preserves the existing filtering behavior. When **per-user-override** is present, the PIX Firewall allows the **permit** or **deny** status from the per-user access-list (if one is downloaded) associated to a user to override the **permit** or **deny** status from the **access-group** command associated access list. Additionally, the following rules are observed:

- At the time a packet arrives, if there is no per-user access list associated with the packet, the interface access list will be applied.
- The per-user access list is governed by the timeout value specified by the **uauth** option of the **timeout** command but it can be overriden by the AAA per-user session timeout value.
- Existing access list log behavior will be the same. For example, if user traffic is denied because of a per-user access list, syslog message 109015 will be logged. If user traffic is permitted, no syslog message is generated. The **log** option in the per-user access-list will have no effect.

Always use the **access-list** command with the **access-group** command.

Note The use of **access-group** command overrides the **conduit** and **outbound** command statements for the specified *interface_name*.

The **no access-group** command unbinds the *access-list* from the interface *interface_name*.

The **show access-group** command displays the current access list bound to the interfaces.

The **clear access-group** command removes all entries from an access list indexed by *access-list*. If *access-list* is not specified, all **access-list** command statements are removed from the configuration.

Examples

The following example shows use of the **access-group** command:

```
static (inside,outside) 209.165.201.3 10.1.1.3
access-list acl_out permit tcp any host 209.165.201.3 eq 80
access-group acl_out in interface outside
```

The **static** command statement provides a global address of 209.165.201.3 for the web server at 10.1.1.3. The **access-list** command statement lets any host access the global address using port 80. The **access-group** command specifies that the **access-list** command statement applies to traffic entering the outside interface.

Related Commands

access-list	Creates an access list, or uses a downloadable access list.

access-list

Create an access list, or use a downloadable access list. (Downloadable access lists are supported for RADIUS servers only).

access-list <acl_name> object-group-search

[**no**] **access-list deny-flow-max** *n*

[**no**] **access-list alert-interval** *secs*

[**no**] **access-list** [*id*] **compiled**

[**no**] **access-list** *id* [**line** *line-num*] **remark** *text*

[**no**] **access-list** *id* [**line** *line-num*] {**deny** | **permit**}{*protocol* | **object-group** *protocol_obj_grp_id* {*source_addr source_mask*} | **object-group** *network_obj_grp_id* [*operator port* [*port*] | **interface** *if_name* | **object-group** *service_obj_grp_id*] {*destination_addr* | *remote_addr*} {*destination_mask* | *remote_mask*} | **object-group** *network_obj_grp_id* [*operator port* [*port*] | **object-group** *service_obj_grp_id*]} [**log** [[**disable** | **default**] | [*level*]]] [**interval** *secs*]]

[**no**] **access-list** *id* [**line** *line-num*] {**deny** | **permit**} **icmp** {*source_addr source_mask*} | **interface** *if_name* | **object-group** *network_obj_grp_id* {*destination_addr* | *remote_addr*} {*destination_mask* | *remote_mask*} | **interface** *if_name* | **object-group** *network_obj_grp_id* [*icmp_type* | **object-group** *icmp_type_obj_grp_id*] [**log** [[**disable** | **default**] | [*level*]]] [**interval** *secs*]]

[**no**] **debug access-list all | standard | turbo**

clear access-list {[*id*] | [*id* **counters**]}

show access-list [[*id*] *source_addr*]

Restricted for use with the **prefix-list** command:
[**no**] **access-list** *id* **deny | permit** {**any** | *prefix mask* | **host** *address*}

Syntax Description

alert-interval *secs*	Specifies the time interval, from 1 to 3600 seconds, for generating syslog message 106101, which alerts you that the firewall has reached a deny flow maximum. In other words, when the deny flow maximum is reached, another 106101 message is generated if has been at least *secs* seconds since the last 106101 message. If this option is not specified, the default interval is 300 seconds.
compiled	When used in conjunction with the **access-list** command, this turns on TurboACL unless the **no** qualifier is used, in which case the command **no access-list** *id* **compiled** turns off TurboACL for that access list. To use TurboACL globally, enter the **access-list compiled** command and to globally turn off TurboACL, enter the **no access-list compiled** command. After TurboACL has been globally configured, individual access lists or groups can have TurboACL enabled or disabled using individual **[no] access-list** *id* **compiled** commands. TurboACL is compiled only if the number of access list elements is greater than or equal to 19.

debug	Outputs access list debugging information to the console.
deny	When used with the **access-group** command, the **deny** option does not allow a packet to traverse the PIX Firewall. By default, PIX Firewall denies all inbound or outbound packets unless you specifically permit access. When used with a **crypto map** command statement, **deny** does not select a packet for IPSec protection. The **deny** option prevents traffic from being protected by IPSec in the context of that particular crypto map entry. In other words, it does not allow the policy as specified in the **crypto map** command statements to be applied to this traffic.
deny-flow-max *n*	Specifies the maximum number of concurrent deny flows that can be created. (Syslog message 106101 is generated when the firewall has reached the maximum number, *n*, of ACL deny flows.) For a firewall with greater than 64 MB Flash memory, the value can be from 1 to 4096, with a default value of 4096. For a firewall with greater than 16 MB Flash memory, the value can be from 1 to 1024, with a default value of 1024. For a firewall with less than or equal to 16 MB Flash memory, the value can be from 1 to 256, with a default value of 256.
destination_addr	IP address of the network or host to which the packet is being sent. Specify a *destination_addr* when the **access-list** command statement is used in conjunction with an **access-group** command statement, or with the **aaa match access-list** command and the **aaa authorization** command. For inbound and outbound connections, *destination_addr* is the address before NAT has been performed.
destination_mask	Netmask bits (mask) to be applied to *destination_addr*, if the destination address is a network mask.
disable	Disables ACL logging for the access control element (ACE), which is an access control list entry.
icmp_type	For non-IPSec use only, permit or deny access to ICMP message types. Refer to Table 3-1 for a list of message types. Omit this option to mean all ICMP types. ICMP message types are not supported for use with IPSec; that is when the **access-list** command is used in conjunction with the **crypto map** command, the *icmp_type* is ignored.
id	Name of an access list. You can use either a name or number.
interface *if_name*	The name of the firewall interface.
interval *secs*	The time interval in seconds, from 1 to 600, at which to generate an 106100 syslog message. The *secs* value is also used as the timeout value for deleting an inactive flow. If this option is not specified, the default interval is 300 seconds for a new access control element (ACE). If an ACE already exists, any interval previously associated with that ACE remains unchanged.
line-num	The line number at which to insert a remark or an access control element (ACE).

log disable | **default** | *level*	When the **log** option is specified, it generates syslog message 106100 for the access list element (ACE) to which it is applied. (Syslog message 106100 is generated for every matching permit or deny ACE flow passing through the firewall.) The first-match flow is cached. Subsequent matches increment the hit count displayed in the **show object-group** command (`hitcnt`) for the ACE, and new 106100 messages will be generated at the end of the interval defined by **interval** *secs* if the hit count for the flow is not zero. The default ACL logging behavior (the **log** keyword not specified) is that if a packet is denied, then message 106023 is generated, and if a packet is permitted, then no syslog message is generated. An optional syslog *level* (0 - 7) may be specified for the generated syslog messages (106100). If no *level* is specified, the default level is 6 (informational) for a new ACE. If the ACE already exists, then its existing log level remains unchanged. If the **log disable** option is specified, access list logging is completely disabled. No syslog message, including message 106023, will be generated. The **log default** option restores the default access list logging behavior.
mask	The netmask.
obj_grp_id	An existing object group.
object-group	Specifies an object group. Refer to the object-group command for information on how to configure object groups.
object-group-search	Use this keyword to specify that access list search is performed on object groups that are contained in access list instead of searching the entire expanded access list. – This mode overrides TurboACL mode (compiled). – When this mode is enabled, TurboACL on this access-list is not allowed. – When this mode is enabled on an access-list, the access-list cannot be used in the **nat** and **crypto** commands.

operator	The *operator* compares the source IP address (*sip*) or destination IP address (*dip*) ports. Possible operands include **lt** for less than, **gt** for greater than, **eq** for equal, **neq** for not equal, and **range** for an inclusive range. Use the **access-list** command the without an operator and port to indicate all ports by default.
	
	For example,
	
	`access-list acl_out permit tcp any host 209.165.201.1`
	
	Use **eq** and a port to permit or deny access to just that port. For example, use **eq ftp** to permit or deny access only to FTP.
	
	`access-list acl_out deny tcp any host 209.165.201.1 eq ftp`
	
	Use **lt** and a port to permit or deny access to all ports less than the port you specify. For example, use **lt 2025** to permit or deny access to the well-known ports (1 to 1024).
	
	`access-list acl_dmz1 permit tcp any host 192.168.1.1 lt 1025`
	
	Use **gt** and a port to permit or deny access to all ports greater than the port you specify. For example, use **gt 42** to permit or deny ports 43 to 65535.
	
	`access-list acl_dmz1 deny udp any host 192.168.1.2 gt 42`
	
	Use **neq** and a port to permit or deny access to every port except the ports that you specify. For example, use **neq 10** to permit or deny ports 1-9 and 11 to 65535.
	
	`access-list acl_dmz1 deny tcp any host 192.168.1.3 neq 10`
	
	Use **range** and a port range to permit or deny access to only those ports named in the range. For example, use **range 10 1024** to permit or deny access only to ports 10 through 1024. All other ports are unaffected. The use of port ranges can dramatically increase the number of IPSec tunnels. For example, if a port range of 5000 to 65535 is specified for a highly dynamic protocol, up to 60,535 tunnels can be created.

permit	When used with the **access-group** command, the **permit** option selects a packet to traverse the PIX Firewall. By default, PIX Firewall denies all inbound or outbound packets unless you specifically permit access. When used with a **crypto map** command statement, **permit** selects a packet for IPSec protection. The **permit** option causes all IP traffic that matches the specified conditions to be protected by IPSec using the policy described by the corresponding **crypto map** command statements.
prefix	The network number. For more information, refer to the **prefix-list** command.
port	Services you permit or deny access to. Specify services by the port that handles it, such as **smtp for port 25**, **www** for port 80, and so on. You can specify ports by either a literal name or a number in the range of 0 to 65535. You can view valid port numbers online at the following website: http://www.iana.org/assignments/port-numbers See "Ports" in Chapter 2, "Using PIX Firewall Commands" for a list of valid port literal names in port ranges; for example, **ftp h323**. You can also specify numbers.
protocol	Name or number of an IP protocol. It can be one of the keywords **icmp**, **ip**, **tcp**, or **udp**, or an integer in the range 1 to 254 representing an IP protocol number. To match any Internet protocol, including ICMP, TCP, and UDP, use the keyword **ip**.

remark *text*	The text of the remark to add before or after an **access-list** command statement, up to 100 characters in length.
remote_addr	IP address of the network or host remote to the PIX Firewall. Specify a *remote_addr* when the **access-list** command statement is used in conjunction with a **crypto access-list** command statement, a **nat 0 access-list** command statement, or a **vpdn group split-tunnel** command statement.
remote_mask	Netmask bits (mask) to be applied to *remote_addr*, if the remote address is a network mask.
source_addr	Address of the network or host from which the packet is being sent. Use this field when an **access-list** command statement is used in conjunction with an **access-group** command statement, or with the **aaa match access-list** command and the **aaa authorization** command.
source_mask	Netmask bits (mask) to be applied to *source_addr*, if the source address is for a network mask.

Defaults

By default, PIX Firewall denies all inbound or outbound packets unless you specifically permit access.

TurboACL is used only if the number of access list elements is greater than or equal to 19.

The default time interval at which to generate syslog message 106100 is 300 seconds.

The default time interval for a deny flow maximum syslog message (106101) is 300 seconds.

The default ACL logging behavior is to generate syslog message 106023 for denied packets.

When the **log** option is specified, the default level for syslog message 106100 is 6 (informational).

Command Modes

Configuration mode.

Usage Guidelines

The **access-list** command lets you specify if an IP address is permitted or denied access to a port or protocol. In this document, one or more **access-list** command statements with the same access list name are referred to as an "access list." Access lists associated with IPSec are known as "crypto access lists."

By default, all **access-list** commands have an implicit **deny** unless you explicitly specify **permit**. In other words, by default, all access in an access list is denied unless you explicitly grant access using a **permit** statement.

Note Do not use the string "multicastACL" following the name of a PIX Firewall interface in an access-list name because this is a reserved keyword used by PIX Device Manager (PDM).

Additionally, you can use the **object-group** command to group access lists like any other network object.

Use the following guidelines for specifying a source or destination address:

- Use a 32-bit quantity in four-part, dotted-decimal format.
- Use the keyword **any** as an abbreviation for an address and mask of 0.0.0.0 0.0.0.0. This keyword is normally not recommended for use with IPSec.
- Use **host** *address* as an abbreviation for a mask of 255.255.255.255.

Use the following guidelines for specifying a network mask:

- Do not specify a mask if the address is for a host; if the destination address is for a host, use the **host** parameter before the address.

  For example:

  ```
  access-list acl_grp permit tcp any host 192.168.1.1
  ```

- If the address is a network address, specify the mask as a 32-bit quantity in four-part, dotted-decimal format. Place zeros in the bit positions you want to ignore.
- Remember that you specify a network mask differently than with the Cisco IOS software **access-list** command. With PIX Firewall, use 255.0.0.0 for a Class A address, 255.255.0.0 for a Class B address, and 255.255.255.0 for a Class C address. If you are using a subnetted network address, use the appropriate network mask.

  For example:

  ```
  access-list acl_grp permit tcp any 209.165.201.0 255.255.255.224
  ```

If appropriate, after you have defined an access list, bind it to an interface using the **access-group** command. For IPSec use, bind it with a **crypto ipsec** command statement. In addition, you can bind an access list with the RADIUS authorization feature (described in the next section).

The **access-list** command supports the **sunrpc** service.

The **show access-list** command lists the **access-list** command statements in the configuration and the hit count of the number of times each element has been matched during an **access-list** command search. Additionally, it displays the number of access list statements in the access list and indicates whether or not the list is configured for TurboACL. (If the list has less than eighteeen access control entries then it is marked to be turbo-configured but is not actually configured for TurboACL until there are 19 or more entries.)

The **show access-list** *source_addr* option filters the show output so that only those access-list elements that match the source IP address (or with **any** as source IP address) are displayed.

The **clear access-list** command removes all **access-list** command statements from the configuration or, if specified, access lists by their *id*. The **clear access-list** *id* **counters** command clears the hit count for the specified access list.

The **no access-list** command removes an **access-list** command from the configuration. If you remove all the **access-list** command statements in an access list, the **no access-list** command also removes the corresponding **access-group** command from the configuration.

Note The **aaa**, **crypto map**, and **icmp** commands make use of the **access-list** command statements.

access-list line *line-num* commands

Use the **access-list** *id* **line** *line-num* command to insert an **access-list** command statement, and the **no access-list** *id* **line** *line-num* command to delete an **access-list** command statement.

Each access control element (ACE) and remark has an associated line number. Line numbers can be used to insert or delete elements at any position in an access list. These numbers are maintained internally in increasing order starting from 1. (For example, in sequence such as 1, 2, 3...) A user can insert a new entry between two consecutive ACEs by choosing the line number of the higher line number ACE.

The line numbers are always maintained in increasing order, with an individual line number for each ACE. However, all ACEs resulting from a single object group **access-list** command statement have a single line number. Consequently, you cannot insert an ACE in the middle of object-group ACEs.

Line numbers are displayed by the **show access-list** command. However, they are not shown in your configuration.

access-list logging commands

The following example shows what happens when an access list log option is enabled. There are some behavior differences among various types of IP traffic because the access check is only applied to those packets which do not have an existing "connection":

```
access-group outside-acl in interface outside
.
.
access-list outside-acl permit ip host 1.1.1.1 any log 7 interval 600
access-list outside-acl permit ip host 2.2.2.2 any
access-list outside-acl deny ip any any log 2
```

The following example illustrates the use of access list based logging in an ICMP context:

1. An inbound ICMP echo request (1.1.1.1 -> 192.168.1.1) arrives on the outside interface.
2. An ACL called **outside-acl** is applied for the access check.
3. The packet is permitted by the first ACE of **outside-acl** which has the **log** option enabled.
4. The log flow (ICMP, 1.1.1.1, 0, 192.168.1.1, 8) has not been cached, so the following syslog message is generated and the log flow is cached:

   ```
   106100: access-list outside-acl permitted icmp outside/1.1.1.1(0) ->
   inside/192.168.1.1(8) hit-cnt 1 (first hit)
   ```

5. Twenty such packets arrive on the outside interface within the next 10 minutes (600 seconds). Because the log flow has been cached, the log flow is located and the hit count of the log flow is incremented for each packet.
6. At the end of 10th minute, the following syslog message is generated and the hit count of the log flow is reset to 0:

   ```
   106100: access-list outside-acl permitted icmp outside/1.1.1.1(0) ->
   inside/192.168.1.1(8) hit-cnt 20 (300-second interval)
   ```

7. No such packets arrive on the outside interface within the next 10 minutes. So the hit count of the log flow remains 0.
8. At the end of 20th minute, the cached flow (ICMP, 1.1.1.1, 0, 192.168.1.1, 8) is deleted because of the 0 hit count.

To disable a log option without having to remove the ACE, use **access-list** *id* **log disable**.

When removing an access control element (ACE) with a log option enabled using a **no access-list** command, it is not necessary to specify all the log options. The ACE is removed as long as its permit or deny rule is used to uniquely identify it. However, the removal of an ACE (with a log option enabled) does not remove the associated cached flows. You must remove the entire access control list (ACL) to remove the cached flows. When a cached flow is flushed due to the removal of an ACL, a syslog message will be generated if the hit count of the flow is non-zero.

The **clear access-list** command removes all the cached flows.

access-list *id* remark command

The **access-list** *id* [**line** *line-num*] **remark** *text* command enables users to include comments (remarks) about entries in any access control list (ACL). You can use remarks to make the ACL easier to scan and interpret. Each remark line is limited to 100 characters.

The ACL remark can be placed before or after an **access-list** command statement, but it should be placed in a consistent position so that it is clear which remark describes which **access-list** command. For example, it would be confusing to have some remarks before the associated **access-list** commands and some remarks after the associated **access-list** commands.

The **no access-list** *id* **line** *line-num* **remark** *text* and **no access-list** *id* **line** *line-num* commands both remove the remark at that line number.

The following are samlpes of possible access list remarks:

```
access-list out-acl remark - ACL for the outside interface
access-list out-acl remark - Allow Joe Smith's group to login
access-list out-acl permit tcp 1.1.1.0 255.255.255.0 server
access-list out-acl remark - Allow Lee White's group to login
access-list out-acl permit tcp 1.1.3.0 255.255.255.0 server
access-list out-acl remark - Deny known hackers
access-list out-acl deny ip host 192.23.56.1 any
access-list out-acl deny ip host 197.1.1.125 any
```

RADIUS Authorization

PIX Firewall allows a RADIUS server to send user group attributes to the PIX Firewall in the RADIUS authentication response message. Additionally, the PIX Firewall allows downloadable access lists from the RADIUS server. For example, you can configure an access list on a Cisco Secure ACS server and download it to the PIX Firewall during RADIUS authorization.

After the PIX Firewall authenticates a user, it can then use the CiscoSecure **acl** attribute returned by the authentication server to identify an access list for a given user group. To maintain consistency, PIX Firewall also provides the same functionality for TACACS+.

To restrict users in a department to three servers and deny everything else, the **access-list** command statements are as follows:

```
access-list eng permit ip any server1 255.255.255.255
access-list eng permit ip any server2 255.255.255.255
access-list eng permit ip any server3 255.255.255.255
access-list eng deny ip any any
```

In this example, the vendor specific attribute string in the CiscoSecure configuration has been set to **acl=eng**. Use this field in the CiscoSecure configuration to identify the **access-list** identification name. The PIX Firewall gets the **acl=***id* from CiscoSecure and extracts the ACL number from the attribute string, which it places in a user's uauth entry. When a user tries to open a connection, PIX Firewall checks the access list in the user's uauth entry, and depending on the permit or deny status of the access list match, permits or denies the connection. When a connection is denied, PIX Firewall generates a corresponding syslog message. If there is no match, then the implicit rule is to deny.

Because the source IP of a given user can vary depending on where they are logging in from, set the source address in the **access-list** command statement to **any**, and the destination address to identify which network services the user is permitted or denied access to. If you want to specify that only users logging in from a given subnet may use the specified services, specify the subnet instead of using **any**.

Note An access list used for RADIUS authorization does not require an **access-group** command to bind the statements to an interface.

There is *not* a **radius** option to the **aaa authorization** command.

Configure the access list specified in Attribute 11 to specify a per-user access list name. Otherwise, remove Attribute 11 from the AAA RADIUS server configuration if no access list is intended for user authentication. If the access list is not configured on the PIX Firewall when the user attempts to login, the login will fail.

For more information on how to use RADIUS server authorization, refer to the *Cisco PIX Firewall and VPN Configuration Guide*, Version 6.2 or higher.

TurboACL

On the PIX Firewall, TurboACL is turned on globally with the command **access-list compiled** (and turned off globally by the command **no access-list compiled**).

The PIX Firewall default mode is TurboACL off (**no access-list compiled**), and TurboACL is active only on access lists with 19 or more entries.

The minimum amount of Flash memory required to run TurboACL is 2.1 MB. If memory allocation fails, the TurboACL lookup tables will not be generated.

Note Use TurboACL only on PIX Firewall platforms that have 16 MB or more of Flash memory. Consequently, TurboACL is not supported on the PIX 501 because it has 8 MB of Flash memory.

If TurboACL is configured, some access control list or access control list group modifications can trigger regeneration of the TurboACL internal configuration. Depending on the extent of TurboACL configuration(s), this could noticeably consume CPU resources. Consequently, we recommend modifying turbo-complied access lists during non-peak system usage hours.

For more information on how to use TurboACL, refer to the *Cisco PIX Firewall and VPN Configuration Guide*, Version 6.2 or higher.

Usage Notes

1. The **clear access-list** command automatically unbinds an access list from a **crypto map** command or interface. The unbinding of an access list from a **crypto map** command can lead to a condition that discards all packets because the **crypto map** command statements referencing the access list are incomplete. To correct the condition, either define other **access-list** command statements to complete the **crypto map** command statements or remove the **crypto map** command statements that pertain to the **access-list** command statement. Refer to the **crypto map** command for more information.
2. Access control lists that are dynamically updated on the PIX Firewall by a AAA server can only be shown using the **show access-list** command. The **write** command does not save or display these updated lists.
3. The **access-list** command operates on a first match basis.
4. If you specify an **access-list** command statement and bind it to an interface with the **access-group** command statement, by default, all traffic inbound to that interface is denied. You must explicitly permit traffic. Note that “inbound” in this context means traffic passing through the interface, rather than the more typical PIX Firewall usage of inbound meaning traffic passing from a lower security level interface to a higher security level interface.
5. Always permit access first and then deny access afterward. If the host entries match, then use a **permit** statement, otherwise use the default **deny** statement. You only need to specify additional **deny** statements if you need to deny specific hosts and permit everyone else.
6. You can view security levels for interfaces with the **show nameif** command.
7. The ICMP message type (*icmp_type*) option is ignored in IPSec applications because the message type cannot be negotiated with ISAKMP.
8. Only one access list can be bound to an interface using the **access-group** command.
9. If you specify the **permit** option in the access list, the PIX Firewall continues to process the packet. If you specify the **deny** option in the access list, PIX Firewall discards the packet and generates the following syslog message.

   ```
   %PIX-4-106019: IP packet from source_addr to destination_addr, protocol protocol
   received from interface interface_name deny by access-group id
   ```

The **access-list** command uses the same syntax as the Cisco IOS software **access-list** command *except* that PIX Firewall uses a subnet mask, whereas Cisco IOS software uses a wildcard mask. (In Cisco IOS software, the mask in this example would be specified with the 0.0.0.255 value.) For example, in the Cisco IOS software **access-list** command, a subnet mask of 0.0.0.255 would be specified as 255.255.255.0 in the PIX Firewall **access-list** command.

10. We recommend that you do not use the **access-list** command with the **conduit** and **outbound** commands. While using these commands together will work, the way in which these commands operate may cause debugging issues because the **conduit** and **outbound** commands operate from one interface to another whereas the **access-list** command used with the **access-group** command applies only to a single interface. If these commands must be used together, PIX Firewall evaluates the **access-list** command before checking the **conduit** and **outbound** commands.
11. Refer to the *Cisco PIX Firewall and VPN Configuration Guide* for a detailed description about using the **access-list** command to provide server access and to restrict outbound user access.
12. Refer to the **aaa-server radius-acctport** and **aaa-server radius-authport** commands to verify or change port settings.

ICMP Message Types

For non-IPSec use only, if you prefer more selective ICMP access, you can specify a single ICMP message type as the last option in this command. Table 3-1 lists possible ICMP types values.

Table 3-1 ICMP Type Literals

ICMP Type	Literal
0	echo-reply
3	unreachable
4	source-quench
5	redirect
6	alternate-address
8	echo
9	router-advertisement
10	router-solicitation
11	time-exceeded
12	parameter-problem
13	timestamp-request
14	timestamp-reply
15	information-request
16	information-reply
17	mask-request
18	mask-reply
31	conversion-error
32	mobile-redirect

If you specify an ICMP message type for use with IPSec, PIX Firewall ignores it.

For example:

```
access-list 10 permit icmp any any echo-reply
```

IPSec is enabled such that a **crypto map** command references the (ACL) *id* for this **access-list** command, then the **echo-repy** ICMP message type is ignored.

Using the access-list Command with IPSec

If an access list is bound to an interface with the **access-group** command, the access list selects which traffic can traverse the PIX Firewall. When bound to a **crypto map** command statement, the access list selects which IP traffic IPSec protects and which traffic IPSec does not protect. For example, access lists can be created to protect all IP traffic between Subnet X and Subnet Y or traffic between Host A and Host B. More information is available in the **crypto map** command section of this guide.

The access lists themselves are not specific to IPSec. It is the **crypto map** command statement referring to 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 the IPSec **crypto map** command statement have these 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 to filter out and discard traffic that IPSec protects.
- 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 **crypto map** command statements with the **ipsec-isakmp** option.) For a peer's initiated IPSec negotiation to be accepted, it must specify a data flow that is permitted by a crypto access list associated with an **ipsec-isakmp** crypto map entry.

You can associate a crypto access list with an interface by defining the corresponding **crypto map** command statement and applying the crypto map set to an interface. Different access lists must be used in different entries of the same crypto map set. However, both inbound and outbound traffic will be evaluated against the same "outbound" IPSec access list. Therefore, the access list's criteria are applied in the forward direction to traffic exiting your PIX Firewall and the reverse direction to traffic entering your PIX Firewall.

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 that specify different IPSec policies.

We recommend that you configure "mirror image" crypto access lists for use by IPSec and that you avoid using the **any** keyword. See the *Cisco PIX Firewall and VPN Configuration Guide* for more information.

If you configure multiple statements for a given crypto access list, in general, the first **permit** statement 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** command statement.

Some services such as FTP require two **access-list** command statements, one for port 10 and another for port 21, to properly encrypt FTP traffic.

Examples

The following example creates a numbered access list that specifies a Class C subnet for the source and a Class C subnet for the destination of IP packets. Because the **access-list** command is referenced in the **crypto map** command statement, PIX Firewall encrypts all IP traffic that is exchanged between the source and destination subnets.

```
access-list 101 permit ip 172.21.3.0 255.255.0.0 172.22.2.0 255.255.0.0
access-group 101 in interface outside
crypto map mymap 10 match address 101
```

The next example only lets an ICMP message type of echo-reply be permitted into the outside interface:

```
access-list acl_out permit icmp any any echo-reply
access-group acl_out interface outside
```

The following example shows how access list entries (ACEs) are numbered by the firewall and how remarks are inserted:

```
pixfirewall(config)# show access-list ac
access-list ac; 2 elements
access-list ac line 1 permit ip any any
access-list ac line 2 permit tcp any any

pixfirewall(config)# access-list ac permit tcp object-group remote object-group locals
pixfirewall(config)# show access-list ac
access-list ac; 3 elements
```

```
access-list ac line 1 permit ip any any
access-list ac line 2 permit tcp any any (
access-list ac line 3 permit tcp object-group remote object-group locals
pixfirewall(config)# access-list ac remark This comment decribes the ACE line 3

pixfirewall(config)# show access-list ac
access-list ac; 3 elements
access-list ac line 1 permit ip any any
access-list ac line 2 permit tcp any any
access-list ac line 3 remark This comment decribes the ACE line 3
access-list ac line 4 permit tcp object-group remote object-group locals

pixfirewall(config)# access-list ac permit tcp 172.16.0.0 255.0.0.0 any
pixfirewall(config)# show access-list ac
access-list ac; 4 elements
access-list ac line 1 permit ip any any
access-list ac line 2 permit tcp any any
access-list ac line 3 remark This comment decribes the ACE line 3
access-list ac line 4 permit tcp object-group remote object-group locals
access-list ac line 5 permit tcp 172.16.0.0 255.0.0.0 any

pixfirewall(config)# no access-list ac permit tcp object-group remote object-group locals
pixfirewall(config)# show access-list ac
access-list ac; 3 elements
access-list ac line 1 permit ip any any
access-list ac line 2 permit tcp any any
access-list ac line 3 remark This comment decribes the ACE line 3
access-list ac line 4 permit tcp 172.16.0.0 255.0.0.0 any
```

The following shows how to remove an access list comment:

```
pixfirewall(config)# access-list ac remark This comment decribes the ACE line 5
pixfirewall(config)# sh access-list ac
access-list ac; 3 elements
access-list ac line 1 permit ip any any
access-list ac line 2 permit tcp any any
access-list ac line 3 remark This comment decribes the ACE line 3
access-list ac line 4 permit tcp 172.16.0.0 255.0.0.0 any
access-list ac line 5 remark This comment decribes the ACE line 5

pixfirewall(config)# no access-list ac remark This comment decribes the ACE line 5
pixfirewall(config)# show access-list ac
access-list ac; 3 elements
access-list ac line 1 permit ip any any line 1
access-list ac line 2 permit tcp any any line 2
access-list ac line 3 remark This comment decribes the ACE line 3
access-list ac line 4 permit tcp 172.16.0.0 255.0.0.0 any line 4
```

The following shows how to insert an access list statement at a specific line number:

```
pixfirewall(config)# show access-list ac
access-list ac; 3 elements
access-list ac line 1 permit ip any any
access-list ac line 2 permit tcp any any
access-list ac line 3 remark This comment decribes the ACE line 3
access-list ac line 4 permit tcp 172.16.0.0255.0.0.0 any

pixfirewall(config)# access-list ac line 4 permit ip 172.16.0.0 255.0.0.0 any
pixfirewall(config)# show access-list ac
access-list ac; 4 elements
access-list ac line 1 permit ip any any
access-list ac line 2 permit tcp any any
access-list ac line 3 remark This comment decribes the ACE line 3
access-list ac line 4 permit ip 172.16.0.0 255.0.0.0 any
```

```
access-list ac line 5 permit tcp 172.16.0.0 255.0.0.0 any
```

The **show access-list** command has the following line of output:

```
access-list cached ACL log flows: total 0, denied 0 (deny-flow-max 4096)
```

which shows the total number of cached ACL log flows (total), the number of cached deny-flows (denied), and the maximum number of allowed deny-flows.

Related Commands

access-group	Binds the access list to an interface.
conduit	(Deprecated command.) Add, delete, or show conduits through the PIX Firewall for incoming connections, superseded by the **access-list** command.
object-group	Defines object groups that you can use to optimize your configuration. Objects such as hosts, protocols, or services can be grouped, and then you can issue a single command using the group name to apply to every item in the group.
outbound/apply	Creates an access list for controlling Internet use.

activation-key

Updates the activation key on your PIX Firewall and checks the activation key running on your PIX Firewall against the activation key stored in the Flash memory of the PIX Firewall.

activation-key *activation-key-four-tuple*

show activation-key

Syntax Description

activation-key	Updates the PIX Firewall activation key unless there is a mismatch between the Flash memory and running PIX Firewall software versions.
activation-key-four-tuple	A four-element hexidecimal string with one space between each element. For example: `0xe02888da 0x4ba7bed6 0xf1c123ae 0xffd8624e` (The leading 0x specfier is optional; all values are assumed to be hexadecimal.)

Defaults

None.

Command Modes

Configuration mode.

Usage Guidelines

Use the **activation-key** *activation-key-four-tuple* command to change the activation key on your PIX Firewall.

Caution Use only an activation key valid for your PIX Firewall software version and platform or your system may not reload after rebooting.

The **activation-key** *activation-key-four-tuple* command output indicates the status of the activation key as follows:

- If the PIX Firewall Flash memory software image version is the same as the running PIX Firewall software version, and the PIX Firewall Flash memory activation key is the same as the running PIX Firewall software activation key, then the **activation-key** command output reads as follows:

```
The flash activation key has been modified.
The flash activation key is now the SAME as the running key.
```

- If the PIX Firewall Flash memory image version is the same as the running PIX Firewall software, and the PIX Firewall Flash memory activation key is different from the running PIX Firewall activation key, then the **activation-key** command output reads as follows:

```
The flash activation key has been modified.
The flash activation key is now DIFFERENT from the running key.
The flash activation key will be used when the unit is reloaded.
```

- If the PIX Firewall Flash memory image version is not the same as the running PIX Firewall software, then the **activation-key** command output reads as follows:

```
The flash image is DIFFERENT from the running image.
The two images must be the same in order to modify the flash activation key.
```

- If the PIX Firewall Flash memory image version is the same as the running PIX Firewall software, and the entered activation key is not valid, then the **activation-key** command output reads as follows:

```
ERROR: The requested key was not saved because it is not valid for this system.
```

- If the PIX Firewall Flash memory activation key is the same as the entered activation key, then the **activation-key** command output reads as follows:

```
The flash activation key has not been modified.
The requested key is the SAME as the flash activation key.
```

The **show activation-key** command output indicates the status of the activation key as follows:

- If the activation key in the PIX Firewall Flash memory is the same as the activation key running on the PIX Firewall, then the **show activation-key** output reads as follows:

```
The flash activation key is the SAME as the running key.
```

- If the activation key in the PIX Firewall Flash memory is the different from the activation key running on the PIX Firewall, then the **show activation-key** output reads as follows:

```
The flash activation key is DIFFERENT from the running key.
The flash activation key takes effect after the next reload.
```

- If the PIX Firewall Flash memory software image version is not the same as the running PIX Firewall software image, then the **show activation-key** output reads as follows:

```
The flash image is DIFFERENT from the running image.
The two images must be the same in order to examine the flash activation key.
```

Usage Notes

1. The PIX Firewall must be rebooted for a new activation key to be enabled.
2. If the PIX Firewall software image is being upgraded to a higher version and the activation key is being updated at the same time, we recommend that you first install the software image upgrade and reboot the PIX Firewall unit, and then update the activation key in the new image and reboot the unit again.
3. If you are downgrading to a lower PIX Firewall software version, we recommend that you ensure that the activation key running on your system is not intended for a higher version before installing the lower version software image. If this is the case, you must first change the activation key to one that is compatible with the the lower version before installing and rebooting. Otherwise, your system may refuse to reload after installation of the new software image.

Examples

The following example shows sample out from the **show activation-key** command:

```
pixfirewall(config)# show activation-key
Serial Number: 480221353 (0x1c9f98a9)

Running Activation Key: 0x36df4255 0x246dc5fc 0x39d2ec4d 0x09f6288f
Licensed Features:
Failover:           Enabled
VPN-DES:            Enabled
VPN-3DES:           Enabled
Maximum Interfaces: 6
Cut-through Proxy:  Enabled
Guards:             Enabled
URL-filtering:      Enabled
Inside Hosts:       Unlimited
Throughput:         Unlimited
IKE peers:          Unlimited

The flash activation key is the SAME as the running key.
```

Related Commands

show version	Displays the PIX Firewall operating information.

alias

Administer overlapping addresses with dual NAT.

[**no**] **alias** [(*if_name*)] *dnat_ip foreign_ip* [*netmask*]

clear alias

show alias

Syntax Description

dnat_ip	An IP address on the internal network that provides an alternate IP address for the external address that is the same as an address on the internal network.
foreign_ip	IP address on the external network that has the same address as a host on the internal network.

if_name	The internal network interface name in which the *foreign_ip* overlaps.
netmask	Network mask applied to both IP addresses. Use 255.255.255.255 for host masks.

Defaults

None.

Command Modes

Configuration mode.

Usage Guidelines

The **alias** command translates one address into another. Use this command to prevent conflicts when you have IP addresses on a network that are the same as those on the Internet or another intranet. You can also use this command to do address translation on a destination address. For example, if a host sends a packet to 209.165.201.1, you can use the **alias** command to redirect traffic to another address, such as, 209.165.201.30.

Note For DNS **fixup** to work properly, **proxy-arp** has to be disabled. If you are using the **alias** command for DNS **fixup**, disable **proxy-arp** with the following command after the **alias** command has been executed:

```
sysopt noproxyarp internal_interface
```

If the **alias** command is used with the **sysopt ipsec pl-compatible** command, a static **route** command statement must be added for each IP address specified in the **alias** command statement.

There must be an A (address) record in the DNS zone file for the "dnat" address in the **alias** command.

Use the **no alias** command to disable a previous **set alias** command statement. Use the **show alias** command to display **alias** command statements in the configuration. Use the **clear alias** command to remove all **alias** commands from the configuration. After changing or removing an **alias** command statement, use the **clear xlate** command.

The **alias** command changes the default behavior of the PIX Firewall in three ways:

- When receiving a packet coming in through the interface identified by *if_name*, destined for the address identified by *dnat_ip*, PIX Firewall sends it to the address identified by *foreign_ip*.
- When receiving a DNS A response, containing the address identified by *foreign_ip*, coming from a lower security interface, and destined for the host behind the inteface identified by *if_name*, PIX Firewall changes *foreign_ip* in the reply to *dnat_ip*. This can be turned off by using the command **sysopt nodnsalias inbound**.
- When receiving a DNS A response, containing the address identified by *dnat_ip*, coming from a DNS server behind the interface, *if_name*, and destined for a host behind the lower security interface, PIX Firewall changes *dnat_ip* address to *foreign_ip*. This can be turned off using the command **sysopt nodnsalias outbound**.

The **alias** command is applied on a per-interface basis, while the **sysopt nodnsalias** changes the behaviour for all interfaces. Also, note that addresses in the zone transfers made across the PIX Firewall, are not changed.

You can specify a net alias by using network addresses for the *foreign_ip* and *dnat_ip* IP addresses. For example, the **alias 192.168.201.0 209.165.201.0 255.255.255.224** command creates aliases for each IP address between 209.165.201.1 and 209.165.201.30.

Note ActiveX blocking does not occur when users access an IP address referenced by the **alias** command. ActiveX blocking is set with the **filter activex** command.

Usage Notes

- To access an **alias** *dnat_ip* address with **static** and **access-list** command statements, specify the *dnat_ip* address in the **access-list** command statement as the address from which traffic is permitted from. The following example illustrates this note.

```
alias (inside) 192.168.201.1 209.165.201.1 255.255.255.255
static (inside,outside) 209.165.201.1 192.168.201.1 netmask 255.255.255.255
access-list acl_out permit tcp host 192.168.201.1 host 209.165.201.1 eq ftp-data
access-group acl_out in interface outside
```

  An alias is specified with the inside address 192.168.201.1 mapping to the foreign address 209.165.201.1.

- You can use the **sysopt nodnsalias** command to disable inbound embedded DNS A record fixups according to aliases that apply to the A record address and outbound replies.

Examples

In the following example, the inside network contains the IP address 209.165.201.29, which on the Internet belongs to example.com. When inside clients try to access example.com, the packets do not go to the PIX Firewall because the client assumes 209.165.201.29 is on the local inside network.

To correct this, use the **alias** command as follows:

```
alias (inside) 192.168.201.0 209.165.201.0 255.255.255.224

show alias
alias 192.168.201.0 209.165.201.0 255.255.255.224
```

When the inside network client 209.165.201.2 connects to example.com, the DNS response from an external DNS server to the internal client's query would be altered by the PIX Firewall to be 192.168.201.29. If the PIX Firewall uses 209.165.200.225 through 209.165.200.254 as the global pool IP addresses, the packet goes to the PIX Firewall with SRC=209.165.201.2 and DST=192.168.201.29. The PIX Firewall translates the address to SRC=209.165.200.254 and DST=209.165.201.29 on the outside.

In the next example, a web server is on the inside at 10.1.1.11 and a **static** command statement was created for it at 209.165.201.11. The source host is on the outside with address 209.165.201.7. A DNS server on the outside has a record for www.example.com as follows:

```
www.example.com. IN A 209.165.201.11
```

The period at the end of the www.example.com. domain name must be included.

The **alias** command follows:

```
alias 10.1.1.11 209.165.201.11 255.255.255.255
```

PIX Firewall doctors the nameserver replies to 10.1.1.11 for inside clients to directly connect to the web server.

The **static** command statement is as follows:

```
static (inside,outside) 209.165.201.11 10.1.1.11
```

The **access-list** command statement you would expect to use follows:

```
access-list acl_grp permit tcp host 209.165.201.7 host 209.165.201.11 eq telnet
```

But with the **alias** command, use this command:

```
access-list acl_grp permit tcp host 209.165.201.11 eq telnet host 209.165.201.7
```

You can test the DNS entry for the host with the following UNIX **nslookup** command:

```
nslookup -type=any www.example.com
```

Related Commands

access-list	Creates an access list, or uses a downloadable access list.
static	Configures a persistent one-to-one address translation rule by mapping a local IP address to a global IP address, also known as Static Port Address Translation (Static PAT).

arp

Configure the Address Resolution Protocol (ARP) cache timeout value, static ARP table entries, or static proxy ARP, and view the ARP cache, status, or timeout value.

[**no**] **arp** *if_name ip mac* [**alias**]

[**no**] **arp timeout** *seconds*

clear arp [**timeout** | **statistics**]

show arp [**timeout** | **statistics**]

Syntax Description

arp	Configure a static ARP mapping (IP-to-physical address binding) for the addresses specified. These entries are not cleared when the ARP persistence timer times out and are automatically stored in the configuration when you use the **write** command to store the configuration.
arp alias	Configure a static proxy ARP mapping (proxied IP-to-physical address binding) for the addresses specified. These entries are not cleared when the ARP persistence timer times out and are automatically stored in the configuration when you use the **write** command to store the configuration.
if_name	The interface name whose ARP table will be changed or viewed. (The interface name itself is specified by the **nameif** command.)
ip	IP address for an ARP table entry.
mac	Hardware MAC address for the ARP table entry; for example, 00e0.1e4e.3d8b.
seconds	Duration that a dynamic ARP entry can exist in the ARP table before being cleared.
statistics	The ARP statistics, including block usage.

Defaults

The default value for the ARP persistence timer is 14,400 seconds (4 hours).

Command Modes

Configuration mode.

Usage Guidelines

The Address Resolution Protocol (ARP) maps an IP address to a MAC address and is defined in RFC 826. Proxy Address Resolution Protocol (proxy ARP) is a variation of the ARP protocol in which an intermediate device (for example, the firewall) sends an ARP response on behalf of an end node to the requesting host. ARP mapping occurs automatically as the firewall processes traffic, however, you can configure the ARP cache timeout value, static ARP table entries, or proxy ARP.

Note Because ARP is a low-level TCP/IP protocol that resolves a node's MAC (physical) address from its IP address (through an ARP request asking the node with a particular IP address to send back its physical address), the presence of entries in the ARP cache indicates that the firewall has network connectivity.

The **arp timeout** command specifies the duration to wait before the ARP table rebuilds itself, automatically updating new host information. This feature is also known as the ARP persistence timer. The **no arp timeout** command resets the ARP persistence timer to its default value. The **show arp timeout** command displays the current timeout value.

The **arp** *if_name ip mac* command adds a static (persistent) entry to the firewall ARP cache. (This matches the behavior of Cisco IOS). For example, you could use the **arp** *if_name ip mac* command to set up a static IP-to-MAC address mapping for hosts on your network. Use the **no arp** *if_name ip mac* command to remove the static ARP mapping.

The **arp** *if_name ip mac* **alias** command configures proxy ARP for the IP and MAC addresses specified. Enable proxy ARP when you want the firewall to respond to ARP requests for another host (determined by the IP address of the host) with the MAC address you specify in the **arp alias** command. Use the **no arp** *if_name ip mac* **alias** command to remove the static proxy ARP mapping.

The **clear arp** command clears all entries in the ARP cache table except for those you configure directly with the **arp** *if_name ip mac* command. Use the **no arp** *if_name ip mac* command to remove these entries. The **show arp** command lists the entries in the ARP table.

The **show arp statistics** command displays the following ARP information:

```
pixfirewall(config)# show arp statistics
        Dropped blocks in ARP: 6
        Maximum Queued blocks: 3
        Queued blocks: 1
        Interface collision ARPs Received: 5
        ARP-defense Gratuitous ARPS sent: 4
        Total ARP retries: 15
        Unresolved hosts: 1
        Maximum Unresolved hosts: 2
```

Examples

The following examples illustrate use of the **arp** and **arp timeout** commands:

```
arp inside 192.168.0.42 00e0.1e4e.2a7c
arp outside 192.168.0.43 00e0.1e4e.3d8b alias
show arp
    outside 192.168.0.43 00e0.1e4e.3d8b alias
    inside 192.168.0.42 00e0.1e4e.2a7c

clear arp inside 192.168.0.42

arp timeout 42
show arp timeout
arp timeout 42 seconds

no arp timeout
show arp timeout
arp timeout 14400 seconds
```

Related Commands		
	sysopt	Changes firewall system options.

auth-prompt

Change the AAA challenge text for through the firewall user sessions. (Configuration mode.)

Configure with the command...	Remove with the command...
auth-prompt [**accept** | **reject** | **prompt**] *string*	**no auth-prompt** [**accept** | **reject** | **prompt**] *string* **clear auth-prompt**

Show command options	Show command output
show auth-prompt	Displays the AAA challenge text.

Syntax Description

accept	If a user authentication via Telnet is accepted, display the prompt *string*.
prompt	The AAA challenge prompt string follows this keyword. This keyword is optional for backward compatibility.
reject	If a user authentication via Telnet is rejected, display the prompt *string*.
string	A string of up to 235 alphanumeric characters or 31 words, limited by whichever maximum is first reached. Special characters should not be used; however, spaces and punctuation characters are permitted. Entering a question mark or pressing the **Enter** key ends the string. (The question mark appears in the string.)

Usage Guidelines

The **auth-prompt** command lets you change the AAA challenge text for HTTP, FTP, and Telnet access through the firewall requiring user authentication from TACACS or RADIUS servers. This text is primarily for cosmetic purposes and displays above the username and password prompts that users view when logging in. If the user authentication occurs from Telnet, you can use the **accept** and **reject** options to display different status prompts to indicate that the authentication attempt is accepted or rejected by the AAA server.

Following is the authentication sequence showing when each **auth-prompt** string is displayed:

1. A user initiates a telnet session from the **inside** interface through the firewall to the **outside** interface.
2. The user receives the **auth-prompt** challenge text, followed by the **username** prompt.
3. The user enters the AAA username/password username and password, or in the formats **aaa_user@outside_use**r and **aaa_pass@outside_pass**.
4. The firewall sends the **aaa_user/aaa_pass** to the TACACS or RADIUS AAA server.
5. If the AAA server authenticates the user, the firewall displays the **auth-prompt accept** text to the user, otherwise the **reject** challenge text is displayed. Authentication of http and ftp sessions displays only the challenge text at the prompt. The **accept** and **reject** text are not displayed.

If you do not use this command, FTP users view `FTP authentication`, HTTP users view `HTTP Authentication`, and challenge text does not appear for Telnet access.

Microsoft Internet Explorer only displays up to 37 characters in an authentication prompt. Netscape Navigator displays up to 120 characters, and Telnet and FTP display up to 235 characters in an authentication prompt.

Examples

The following example shows how to set the authentication prompt and how users view the prompt:

```
auth-prompt XYZ Company Firewall Access
```

After this string is added to the configuration, users view the following:

```
Example.com Company Firewall Access
User Name:
Password:
```

The **prompt** keyword can be included or omitted.

For example:

```
auth-prompt prompt Hello There!
```

This command statement is the same as the following:

```
auth-prompt Hello There!
```

Related Commands

aaa authentication	Enables, disables, or displays LOCAL, TACACS+, or RADIUS user authentication on a server designated by the **aaa-server** command, or for PDM user authentication.

auto-update

Specifies how often to poll an Auto Update Server.

[**no**] **auto-update device-id hardware-serial | hostname | ipaddress** [*if_name*] **| mac-address** [*if_name*] **| string** *text*

[**no**] **auto-update poll-period** *poll_period* [*retry_count* [*retry_period*]]

clear auto-update

[**no**] **auto-update server** *url* [*verify_certificate*]

[**no**] **auto-update timeout** *period*

clear auto-update

show auto-update

Syntax Description

device-id	The device ID of the PIX Firewall.
hardware-serial	Specifies to use the hardware serial number of the PIX Firewall to uniquely identify the device.
hostname	Specifies to use the host name of the PIX Firewall to uniquely identify the device.

if_name	Specifies the interface to use (with its corresponding IP or MAC address) to uniquely identify the device.
ipaddress	Specifies to use the IP address of the specified PIX Firewall interface to uniquely identify the firewall.
mac-address	Specifies to use the MAC address of the specified PIX Firewall interface to uniquely identify the firewall.
period	Specifies how long to attempt to contact the Auto Update Server, after the last successful contact, before stopping all traffic passing through the firewall.
poll_period	Specifies how often, in minutes, to poll an Auto Update Server. The default is 720 minutes (12 hours).
retry_count	Specifies how many times to try reconnecting to the Auto Update Server if the first attempt fails. The default is 0.
retry_period	Specifies how long to wait, in minutes, between connection attempts. The default is 5 minutes and the valid range of values is from 1 to 35791.
text	Specifies the text string to uniquely identify the device to the Auto Update Server.
url	Specifies the location of the Auto Update Server using the following syntax: **http**[**s**]**:**[[*user:password*@] *location* [:*port*]] **/** *pathname* See the **copy** command for variable descriptions.
verify_certificate	Specifies to verify the certificate returned by the Auto Update Server.

Defaults

The default poll period is 720 minutes (12 hours).

The default number of times to try reconnecting to the Auto Update Server if the first attempt fails is 0.

The default period to wait between connection attempts is 5 minutes.

Command Modes

Configuration mode.

Usage Guidelines

The **clear auto-update** command removes the entire auto-update configuration.

The **auto-update poll-period** command specifies how often to poll the Auto Update Server for configuration or software image updates. The **no auto-update poll-period** command resets the poll period to the default.

The **auto-update server** command specifies the URL of the Auto Update Server. Only one server can be configured. The **no auto-update server** command disables polling for auto-update updates (by terminating the auto-update daemon).

The **auto-update timeout** command is used to stop all new connections to the PIX Firewall if the Auto Update Server has not been contacted for *period* minutes. This can be used to ensure that the PIX Firewall has the most recent image and configuration.

The **show auto-update** command displays the Auto Update Server, poll time, and timeout period.

Examples

The **show auto-update** command displays the Auto Update Server, poll time, and timeout period. The following is sample output from the command:

```
show auto-update
Server: https://10.0.1.15/autoupdate/AutoUpdateServlet
Poll period: 1 minutes, retry count: 0, retry period: 5 minutes
Timeout: none
Device ID: string [device1]
Next poll in 0.13 minutes
Last poll: 23:43:33 UTC Fri Jun 7 2002
```

The format of the URL, `/autoupdate/AutoUpdateServlet`, is the standard URL format on the Auto Update Server. The port 443 (the default port for HTTPS) can be omitted because it is the default setting.

Related Commands

copy	Changes software images without requiring access to the TFTP monitor mode.

banner

Configures the session, login, or message-of-the-day banner.

banner {**exec | login | motd**} *text*

no banner {**exec | login | motd**} [*text*]

show banner [{**exec | login | motd**}]

clear banner

Syntax Description

exec	Configures the system to display a banner before displaying the enable prompt.
login	Configures the system to display a banner before the password login prompt when accessing the firewall using telnet.
motd	Configures the system to display a message-of-the-day banner.
text	The line of message text to be displayed in the firewall CLI. Subsequent *text* entries are added to the end of an existing banner unless the banner is cleared first. The tokens $(domain) and $(hostname) are replaced with the host name and domain name of the firewall.

Defaults

The default is no login, session, or message-of-the-day banner.

Command Modes

The **banner** command is available in configuration mode.

The **show banner** command is available in privileged mode.

Usage Guidelines

The **banner** command configures a banner to display for the option specified. The *text* string consists of all characters following the first whitespace (space) until the end of the line (carriage return or LF). Spaces in the text are preserved. However, tabs cannot be entered through the CLI.

Multiple lines in a banner are handled by entering a new banner command for each line you wish to add. Each line is then appended to the end of the existing banner. If the text is empty, then a carriage return (CR) will be added to the banner. There is no limit on the length of a banner other than RAM and Flash memory limits.

When accessing the firewall through Telnet or SSH, the session closes if there is not enough system memory available to process the banner messages or if a TCP write error occurs in attempting to display the banner messages.

To replace a banner, use the **no banner** command before adding the new lines. The **no banner** {**exec | login | motd**} command removes all the lines for the banner option specified. The **no banner** command does not selectively delete text strings, so any *text* entered at the end of the **no banner** command is ignored.

The **clear banner** command removes all the banners.

The **show banner** {**motd | exec | login**} command displays the specified banner option and all the lines configured for it. If a banner option is not specified, then all the banners are displayed.

Examples

The following example shows how to configure the **motd**, **exec**, and **login** banners:

```
pixfirewall(config)# banner motd Think on These Things
pixfirewall(config)# banner exec Enter your password carefully
pixfirewall(config)# banner login Enter your password to log in
pixfirewall(config)# show banner
exec:
Enter your password carefully

login:
Enter your password to log in

motd:
Think on These Things
```

The following example shows how to add a second line to a banner:

```
pixfirewall(config)# banner motd and Enjoy Today
pixfirewall(config)# show banner motd
Think on These Things
and Enjoy Today
```

Related Commands

login	Specifies to log in as a particular user.
password	Sets the password for Telnet access to the PIX Firewall console.

CHAPTER 4

C Commands

ca

Configure the PIX Firewall to interoperate with a certification authority (CA).

ca authenticate *ca_nickname* [*fingerprint*]

[**no**] **ca configure** *ca_nickname* **ca** | **ra** *retry_period retry_count* [**crloptional**]

[**no**] **ca crl request** *ca_nickname*

[**no**] **ca enroll** *ca_nickname challenge_password* [**serial**] [**ipaddress**]

ca generate rsa {**key** | **specialkey**} *key_modulus_size*

[**no**] **ca identity** *ca_nickname* [*ca_ipaddress*| *hostname* [**:***ca_script_location*] [*ldap_ip address*| *hostname*]]

[**no**] **ca save all**

[**no**] **ca subject-name** *ca_nickname X.500_string*

[**no**] **ca verifycertdn** *X.500_string*

ca zeroize rsa [*keypair_name*]

show ca certificate

show ca crl

show ca configure

show ca identity

show ca mypubkey rsa

show ca subject-name

show ca verifycertdn

Syntax Description

ca_ipaddress	The CA's IP address.
ca_nickname	The name of the certification authority (CA). Enter any string that you desire. (If you previously declared the CA and just want to update its characteristics, specify the name you previously created.) The CA might require a particular name, such as its domain name. Currently, the PIX Firewall supports only one CA at a time.
ca | **ra**	Indicates whether to contact the CA or registration authority (RA) when using the **ca configure** command. Some CA systems provide an RA, which the PIX Firewall contacts instead of the CA.
:*ca_script_location*	The default location and script on the CA server is /cgi-bin/pkiclient.exe. If the CA administrator has not put the CGI script in this location, provide the location and the name of the script in the **ca identity** command. A PIX Firewall uses a subset of the HTTP protocol to contact the CA, and so it must identify a particular cgi-bin script to handle CA requests.
challenge_password	A required password that gives the CA administrator some authentication when a user calls to ask for a certificate to be revoked. It can be up to 80 characters in length.
crloptional	Allows other peers' certificates be accepted by your PIX Firewall even if the appropriate certificate revocation list (CRL) is not accessible to your PIX Firewall. The default is without the **crloptional** option.
fingerprint	A key consisting of alphanumeric characters the PIX Firewall uses to authenticate the CA's certificate.
hostname	The host name.
ipaddress	Return the PIX Firewall unit's IP address in the certificate.
key	Specifies that one general-purpose RSA key pair will be generated.
key_modulus_size	The size of the key modulus, which is between 512 and 2048 bits. Choosing a size greater than 1024 bits may cause key generation to take a few minutes.
ldap_ipaddress	The IP address of the Lightweight Directory Access Protocol (LDAP) server. By default, querying of a certificate or a CRL is done via Cisco's PKI protocol. If the CA supports LDAP, query functions may also use LDAP.
retry_count	Specify how many times the PIX Firewall will resend a certificate request when it does not receive a certificate from the CA from the previous request. Specify from 1 to 100. The default is 0, which indicates that there is no limit to the number of times the PIX Firewall should contact the CA to obtain a pending certificate.
retry_period	Specify the number of minutes the PIX Firewall waits before resending a certificate request to the CA when it does not receive a response from the CA to its previous request. Specify from 1 to 60 minutes. By default, the PIX Firewall retries every 1 minute.
serial	Return the PIX Firewall unit's serial number in the certificate.
specialkey	This specifies that two special-purpose RSA key pairs will be generated instead of one general-purpose key.
subject-name	Configures the device certificate request with the specified subject name.

verifycertdn	Verifies the certificate's Distinguished Name (DN) and acts as a subject name filter, based on the *X.500_string*. If the subject name of the peer certificate matches the *X.500_string*, then it is filtered out and ISAKMP negotiation fails.
X.500_string	Specify per RFC1779. The entered string will be the Distinguished Name (DN) sent.

Defaults

The *retry_count* default is 0.

Command Modes

Configuration mode.

Usage Guidelines

The sections that follow describe each **ca** command.

The PIX Firewall currently supports the CA servers from VeriSign, Entrust, Baltimore Technologies, and Microsoft. Refer to the *Cisco PIX Firewall and VPN Configuration Guide* for a list of specific CA server versions the PIX Firewall supports.

The lifetime of a certificate and the certificate revocation list (CRL) is checked in UTC, which is the same as GMT. Set the PIX Firewall clock to UTC to ensure that CRL checking works correctly. Use the **clock** command to set the PIX Firewall clock.

The PIX Firewall authenticates the entity certificate (the device certificate). The PIX Firewall assumes the entity certificate is issued by the same trusted point or root (the CA server). As a result, they should have the same root certificate (issuer certificate). Therefore, the PIX Firewall assumes the entity exchanges the entity certificate only, and cannot process a certificate chain that includes both the entity and root certificates.

ca authenticate

The **ca authenticate** command allows the PIX Firewall to authenticate its certification authority (CA) by obtaining the CA's self-signed certificate, which contains the CA's public key.

To authenticate a peer's certificate(s), a PIX Firewall must obtain the CA certificate containing the CA public key. Because the CA certificate is a self-signed certificate, the key should be authenticated manually by contacting the CA administrator. You are given the choice of authenticating the public key in that certificate by including within the **ca authenticate** command the key's fingerprint, which is retrieved in an out-of-band process. The PIX Firewall will discard the received CA certificate and generate an error message, if the fingerprint you specified is different from the received one. You can also simply compare the two fingerprints without having to enter the key within the command.

If you are using RA mode (within th**e ca configure** command), when you issue the **ca authenticate** command, the RA signing and encryption certificates will be returned from the CA, as well as the CA certificate.

The **ca authenticate** command is not saved to the PIX Firewall configuration. However, the public keys embedded in the received CA (and RA) certificates are saved in the configuration as part of the RSA public key record (called the "RSA public key chain"). To save the public keys permanently to Flash memory, use the **ca save all** command. To view the CA's certificate, use the **show ca certificate** command.

Note If the CA does not respond by a timeout period after this command is issued, the terminal control will be returned so it will not be tied up. If this happens, you must re-enter the command.

ca configure

The **ca configure** command is used to specify the communication parameters between the PIX Firewall and the CA.

Use the **no ca configure** command to reset each of the communication parameters to the default value. If you want to show the current settings stored in RAM, use the **show ca configure** command.

The following example indicates that *myca* is the name of the CA and the CA will be contacted rather than the RA. It also indicates that the PIX Firewall will wait 5 minutes before sending another certificate request, if it does not receive a response, and will resend a total of 15 times before dropping its request. If the CRL is not accessible, **crloptional** tells the PIX Firewall to accept other peer's certificates.

```
ca configure myca ca 5 15 crloptional
```

ca crl request

The **ca crl request** command allows the PIX Firewall to obtain an updated CRL from the CA at any time. The **no ca crl command deletes the CRL within the** PIX Firewall**.**

A CRL lists all the network's devices' certificates that have been revoked. The PIX Firewall will not accept revoked certificates; therefore, any peer with a revoked certificate cannot exchange IPSec traffic with your PIX Firewall.

The first time your PIX Firewall receives a certificate from a peer, it will download a CRL from the CA. Your PIX Firewall then checks the CRL to make sure the peer's certificate has not been revoked. (If the certificate appears on the CRL, it will not accept the certificate and will not authenticate the peer.)

A CRL can be reused with subsequent certificates until the CRL expires. When the CRL does expire, the PIX Firewall automatically updates it by downloading a new CRL and replaces the expired CRL with the new CRL.

If your PIX Firewall has a CRL which has not yet expired, but you suspect that the CRL's contents are out of date, use the **ca crl request** command to request that the latest CRL be immediately downloaded to replace the old CRL.

The **ca crl request** command is not saved with the PIX Firewall configuration between reloads.

The following example indicates the PIX Firewall will obtain an updated CRL from the CA with the name myca:

```
ca crl request myca
```

The **show ca crl** command lets you know whether there is a CRL in RAM, and where and when the CRL is downloaded.

The following is sample output from the **show ca crl** command. See Table 4-2 for descriptions of the strings within the following sample output.

```
show ca crl

CRL:
    CRL Issuer Name:
        CN = MSCA, OU = Cisco, O = VSEC, L = San Jose, ST = CA, C = US, EA
=<16> username@example.com
    LastUpdate:17:07:40 Jul 11 2000

    NextUpdate:05:27:40 Jul 19 2000
```

ca enroll

The **ca enroll** command is used to send an enrollment request to the CA requesting a certificate for all of your PIX Firewall unit's key pairs. This is also known as "enrolling" with the CA. (Technically, enrolling and obtaining certificates are two separate events, but they both occur when this command is issued.)

Your PIX Firewall needs a signed certificate from the CA for each of its RSA key pairs; if you previously generated general purpose keys, the **ca enroll** command will obtain one certificate corresponding to the one general purpose RSA key pair. If you previously generated special usage keys, this command will obtain two certificates corresponding to each of the special usage RSA key pairs.

If you already have a certificate for your keys, you will be unable to complete this command; instead, you will be prompted to remove the existing certificate first.

The **ca enroll** command is not saved with the PIX Firewall configuration between reloads. To verify if the enrollment process succeeded and to display the PIX Firewall unit's certificate, use the **show ca certificate** command. If you want to cancel the current enrollment request, use the **no ca enroll** command.

The required challenge password is necessary in the event that you need to revoke your PIX Firewall unit's certificate(s). When you ask the CA administrator to revoke your certificate, you must supply this challenge password as a protection against fraudulent or mistaken revocation requests.

Note This password is not stored anywhere, so you must remember this password.

If you lose the password, the CA administrator may still be able to revoke the PIX Firewall's certificate, but will require further manual authentication of the PIX Firewall administrator identity.

The PIX Firewall unit's serial number is optional. If you provide the **serial** option, the serial number will be included in the obtained certificate. The serial number is not used by IPSec or IKE but may be used by the CA to either authenticate certificates or to later associate a certificate with a particular device. Ask your CA administrator if serial numbers should be included in the certificate. If you are in doubt, specify the **serial** option.

The PIX Firewall unit's IP address is optional. If you provide the **ipaddress** option, the IP address will be included in the obtained certificate. Normally, you would not include the **ipaddress** option because the IP address binds the certificate more tightly to a specific entity. Also, if the PIX Firewall is moved, you would need to issue a new certificate.

Note When configuring ISAKMP for certificate-based authentication, it is important to match the ISAKMP identity type with the certificate type. The **ca enroll** command used to acquire certificates will, by default, get a certificate with the identity based on host name. The default identity type for the **isakmp identity** command is based on address instead of host name. You can reconcile this disparity of identity types by using the **isakmp identity address** command. See the **isakmp** command for information about the **isakmp identity address** command.

The following example indicates that the PIX Firewall will send an enrollment request to the CA myca.example.com. The password 1234567890 is specified, as well as a request for the PIX Firewall unit's serial number to be embedded in the certificate.

```
ca enroll myca.example.com 1234567890 serial
```

ca generate rsa

The **ca generate rsa** command generates RSA key pairs for your PIX Firewall. RSA keys are generated in pairs—one public RSA key and one private RSA key. If your PIX Firewall already has RSA keys when you issue this command, you will be warned and prompted to replace the existing keys with new keys.

Note Before issuing this command, make sure your PIX Firewall has a host name and domain name configured (using the **hostname** and **domain-name** commands). You will be unable to complete the **ca generate rsa** command without a host name and domain name.

The **ca generate rsa** command is not saved in the PIX Firewall configuration. However, the keys generated by this command are saved in the persistent data file in Flash memory, which is never displayed to the user or backed up to another device.

In this example, one general-purpose RSA key pair is to be generated. The selected size of the key modulus is 2048.

```
ca generate rsa key 2048
```

Note You cannot generate both special usage and general purpose keys; you can only generate one or the other.

ca identity

The **ca identity** command declares the CA that your PIX Firewall will use. Currently, PIX Firewall supports one CA at one time. The **no ca identity** command removes the **ca identity** command from the configuration and deletes all certificates issued by the specified CA and CRLs. The **show ca identity** command shows the current settings stored in RAM.

The PIX Firewall uses a subset of the HTTP protocol to contact the CA, and so must identify a particular cgi-bin script to handle CA requests. The default location and script on the CA server is /cgi-bin/pkiclient.exe. If the CA administrator has not put the CGI script in the previously listed location, include the location and the name of the script within the **ca identity** command statement.

By default, querying of a certificate or a CRL is done via Cisco's PKI protocol. If the CA supports Lightweight Directory Access Protocol (LDAP), query functions may use LDAP as well. The IP address of the LDAP server must be included within the **ca identity** command statement.

The following example indicates that the CA myca.example.com is declared as the PIX Firewall unit's supported CA. The CA's IP address of 205.139.94.231 is provided.

```
ca identity myca.example.com 205.139.94.231
```

ca save all

The **ca save all** commands lets you save the PIX Firewall unit's RSA key pairs, the CA, RA and PIX Firewall unit's certificates, and the CA's CRLs in the persistent data file in Flash memory between reloads. The **no ca save** command removes the saved data from PIX Firewall unit's Flash memory.

The **ca save** command itself is not saved with the PIX Firewall configuration between reloads.

To view the current status of requested certificates, and relevant information of received certificates, such as CA and RA certificates, use the **show ca certificate** command. Because the certificates contain no sensitive data, any user can issue this **show** command.

ca subject-name *ca_nickname X.500_string*

The **ca subject-name** *ca_nickname X.500_string* command is a certificate enrollment enhancement that supports X.500 directory names.

When the **ca subject-name** *ca_nickname X.500_string* command is configured, the firewall enrolls the device certificate with the subject Distinguished Name (DN) that is specified in the *X.500_string*, using RFC 1779 format. The supported DN attributes are listed in Table 4-1

Table 4-1 Supported Distinguished Name attributes.

Attribute	Description
`ou`	OrganizationalUnitName
`o`	OrganizationName
`st`	StateOrProvinceName
`c`	CountryName
`ea`	Email address (a non-RFC 1779 format attribute)

For more information on RFC 1779, refer to http://www.ietf.org/rfc/rfc1779.txt.

PIX Firewall software Version 6.3 supports X.509 (certificate support) on the VPN client. Cisco IOS software, the VPN 3000 concentrator, and the PIX Firewall look for the correct VPN group (mode config group) according to the `ou` attribute. (The *ou* attribute is part of the subject DN of the device certificate when the Easy VPN client negotiates the RSA signature.) For example,

```
ca subject-name myca ou=my_department, o=my_org, st=CA, c=US
```

where *my_department* is the VPN group.

Note If the *X.500_string* is being using to communicate between a Cisco VPN 3000 headend and the firewall, the VPN 3000 headend must not be configured to use DNS names for its backup servers. Instead, the backup servers must be specified by their IP addresses.

ca verifycertdn *X.500_string*

The **ca verifycertdn** *X.500_string* command verifies the certificate's Distinguished Name (DN) and acts as a subject name filter, based on the *X.500_string*. If the subject name of the peer certificate matches the *X.500_string*, then it is filtered out and ISAKMP negotiation fails.

ca zeroize rsa

The **ca zeroize rsa** command deletes all RSA keys that were previously generated by your PIX Firewall. If you issue this command, you must also perform two additional tasks. Perform these tasks in the following order:

1. Use the **no ca identity** command to manually remove the PIX Firewall unit's certificates from the configuration**.** This will delete all the certificates issued by the CA.
2. Ask the CA administrator to revoke your PIX Firewall unit's certificates at the CA. Supply the challenge password you created when you originally obtained the PIX Firewall unit's certificates using the **crypto ca enroll** command.

To delete a specific RSA key pair, specify the name of the RSA key you want to delete using the option *keypair_name* within the **ca zeroize rsa** command statement.

Note You may have more than one pair of RSA keys due to SSH. See the **ssh** command in Chapter 8, "S Commands" for more information.

show ca commands

The **show ca certificate** command displays the CA Server's subject name, CRL distribution point (where the PIX Firewall will obtain the CRL), and lifetime of both the CA server's root certificate and the PIX Firewall's certificates.

The following is sample output from the **show ca certificate** command. The CA certificate stems from a Microsoft CA server previously generated for this PIX Firewall.

```
show ca certificate

RA Signature Certificate
  Status:Available
  Certificate Serial Number:6106e08a000000000005
  Key Usage:Signature
    CN = SCEP
     OU = VSEC
     O = Cisco
     L = San Jose
     ST = CA
     C = US
     EA =<16> username@example.com
  Validity Date:
    start date:17:17:09 Jul 11 2000

    end   date:17:27:09 Jul 11 2001


Certificate
  Status:Available
  Certificate Serial Number:1f80655400000000000a
  Key Usage:General Purpose
  Subject Name
    Name:pixfirewall.example.com
  Validity Date:
    start date:20:06:23 Jul 17 2000

    end   date:20:16:23 Jul 17 2001


CA Certificate
  Status:Available
  Certificate Serial Number:25b81813efe58fb34726eec44ae82365
  Key Usage:Signature
    CN = MSCA
     OU = Cisco
     O = VSEC
     L = San Jose
     ST = CA
     C = US
     EA =<16> username@example.com
  Validity Date:
    start date:17:07:34 Jul 11 2000
RA KeyEncipher Certificate
  Status:Available
  Certificate Serial Number:6106e24c000000000006
  Key Usage:Encryption
    CN = SCEP
```

```
    OU = VSEC
    O = Cisco
    L = San Jose
    ST = CA
    C = US
    EA =<16> username@example.com
  Validity Date:
    start date:17:17:10 Jul 11 2000

    end   date:17:27:10 Jul 11 01
```

Table 4-2 describes strings within the **show ca certificate** command sample output.

Table 4-2 show ca certificate command Output Strings

Sample Output String	Description
CN	common name
C	country
EA	E-mail address
L	locality
ST	state or province
O	organization name
OU	organizational unit name
DC	domain component

The **show ca crl** command displays whether there is a certificate revocation list (CRL) in the PIX Firewall RAM, and where and when the CRL downloaded.

The **show ca configure** command displays the current communication parameter settings stored in the PIX Firewall RAM.

The **show ca identity** command displays the the current certification authority (CA) settings stored in RAM.

The **show ca mypubkey rsa** command displays the PIX Firewall unit's public keys in a DER/BER encoded PKCS#1 representation.

The following is sample output from the **show ca mypubkey rsa** command. Special usage RSA keys were previously generated for this PIX Firewall using the **ca generate rsa** command.

```
show ca mypubkey rsa

% Key pair was generated at: 15:34:55 Aug 05 1999

Key name: pixfirewall.example.com
 Usage: Signature Key
 Key Data:
  305c300d 06092a86 4886f70d 01010105 00034b00 30480241 00c31f4a ad32f60d
  6e7ed9a2 32883ca9 319a4b30 e7470888 87732e83 c909fb17 fb5cae70 3de738cf
  6e2fd12c 5b3ffa98 8c5adc59 1ec84d78 90bdb53f 2218cfe7 3f020301 0001
% Key pair was generated at: 15:34:55 Aug 05 1999

Key name: pixfirewall.example.com
 Usage: Encryption Key
 Key Data:
  305c300d 06092a86 4886f70d 01010105 00034b00 30480241 00d8a6ac cc64e57a
  48dfb2c1 234661c7 76380bd5 72ae62f7 1706bdab 0eedd0b5 2e5feef0 76319d98
```

```
908f50b4 85a291de 247b6711 59b30026 453bfa3c 45234991 5d020301 0001
```

Examples

In the following example, a request for the CA's certificate was sent to the CA. The fingerprint was not included in the command. The CA sends its certificate and the PIX Firewall prompts for verification of the CA's certificate by checking the CA certificate's fingerprint. Using the fingerprint associated with the CA's certificate retrieved in some out-of-band process from a CA administrator, compare the two fingerprints. If both fingerprints match, then the certificate is considered valid.

```
ca authenticate myca
Certificate has the following attributes:
Fingerprint: 0123 4567 89AB CDEF 0123
```

The following example shows the error message. This time, the fingerprint is included in the command. The two fingerprints do not match, and therefore the certificate is not valid.

```
ca authenticate myca 0123456789ABCDEF0123
Certificate has the following attributes:
Fingerprint: 0123 4567 89AB CDEF 5432
%Error in verifying the received fingerprint. Type help or '?' for a list of
available commands.
```

ca generate rsa key

The **ca generate rsa** command generates RSA key pairs for your PIX Firewall. RSA keys are generated in pairs—one public RSA key and one private RSA key.

ca generate rsa key *modulus*

Syntax Description

ca generate rsa key	Generates an RSA key for the PIX Firewall.
modulus	Defines the modulus used to generate the RSA key. This is a size measured in bits. You can specify a modulus between 512, 768, 1024, and 2048.

Note Before issuing this command, make sure your PIX Firewall host name and domain name have been configured (using the **hostname** and **domain-name** commands). If a domain name is not configured, the PIX Firewall uses a default domain of *ciscopix.com*.

Defaults

RSA key modulus default (during PDM setup) is 768. The default domain is *ciscopix.com*.

Command Modes

Configuration mode.

Usage Guidelines

If your PIX Firewall already has RSA keys when you issue this command, you are warned and prompted to replace the existing keys with new keys.

Note The larger the key modulus size you specify, the longer it takes to generate an RSA. We recommend a default value of 768.

PDM uses the Secure Sockets Layer (SSL) communications protocol to communicate with the PIX Firewall.

SSL uses the private key generated with the **ca generate rsa** command. For a certificate, SSL uses the key obtained from a certification authority (CA). If that does not exist, it uses the PIX Firewall self-signed certificate created when the RSA key pair was generated.

If there is no RSA key pair when an SSL session is initiated, the PIX Firewall creates a default RSA key pair using a key modulus of 768.

The **ca generate rsa** command is not saved in the PIX Firewall configuration. However, the keys generated by this command are saved in a persistent data file in Flash memory, which can be viewed with the **show ca my rsa key** command.

Examples

The following example demonstrates how one general purpose RSA key pair is generated. The selected size of the key modulus is 1024.

```
router(config) ca generate rsa key 1024
Key name:pixfirewall.cisco.com
 Usage:General Purpose Key
 Key Data:
  30819f30 0d06092a 864886f7 0d010101 05000381 8d003081 89028181 00c8ed4c
  9f5e0b52 aea931df 04db2872 5c4c0afd 9bd0920b 5e30de82 63d834ac f2e1db1f
  1047481a 17be5a01 851835f6 18af8e22 45304d53 12584b9c 2f48fad5 31e1be5a
  bb2ddc46 2841b63b f92cb3f9 8de7cb01 d7ea4057 7bb44b4c a64a9cf0 efaacd42
  e291e4ea 67efbf6c 90348b75 320d7fd3 c573037a ddb2dde8 00df782c 39020301 0001
```

capture

Enables packet capture capabilities for packet sniffing and network fault isolation.

capture *capture_name* [**access-list** *acl_name*][**buffer** *bytes*] [**ethernet-type** *type*][**interface** *name*] [**packet-length** *bytes*] [**circular-buffer**]

no capture *capture_name* [**access-list** [*acl_name*]] [**interface** *name*] [**circular-buffer**]

clear capture *capture_name*

show capture [*capture_name*] [**access-list** *acl_name*] [**detail**] [**dump**]

Syntax Description

access-list	Selects packets based on IP or higher fields. By default, all IP packets are matched.
acl_name	The access list *id.*
buffer	Defines the buffer size used to store the packet. The default size is 512 KB. Once the buffer is full, packet capture stops.
bytes	The number of bytes (b) to allocate.
capture_name	A name to uniquely identify the packet capture.
circular-buffer	Overwrites the buffer, starting from the beginning, when the buffer is full.

detail	Shows additional protocol information for each packet.
dump	Shows a hexidecimal dump of the packet transported over the data link transport. (However, the MAC information is not shown in the hex dump.)
ethernet-type	Selects packets based on the Ethernet type. An exception is the 802.1Q or VLAN type. The 802.1Q tag is automatically skipped and the inner Ethernet type is used for matching. By default, all Ethernet types are accepted.
interface	The interface for packet capture.
name	The name of the interface on which to use packet capture.
packet-length	Sets the maximum number of bytes of each packet to store in the capture buffer. By default, the maximum is 68 bytes.
type	An Ethernet type to exclude from capture. The default is **0**, so you can restore the default at any time by setting *type* to **0**.

Defaults

The default *type* is 0.

Command Modes

Configuration mode.

Usage Guidelines

To enable packet capturing, attach the capture to an interface with the *interface* option. Multiple interface statements attach the capture to multiple interfaces.

If the buffer contents are copied to a TFTP server in ASCII format, then only the headers can be seen. The details and hex dump of the packets can not be seen. To see the details and hex dump, transfer the buffer in PCAP format and then read with TCPDUMP or Ethereal using the options to show the detail and hex dump of the packets.

The **ethernet-type** and **access-list** options select the packets to store in the buffer. A packet must pass both the Ethernet and access list filters before the packet is stored in the capture buffer.

The **capture** *capture_name* **circular-buffer** command enables the capture buffer to overwrite itself, starting from the beginning, when the capture buffer is full.

Enter the **no capture** command with either the **access-list** or **interface** option unless you want to clear the capture itself. Entering **no capture** without options deletes the capture. If the **access-list** option is specified, the access list is removed from the capture and the capture is preserved. If the **interface** option is specified, the capture is detached from the specified interface and the capture is preserved.

To clear the capture buffer, use the **clear capture** *capture_name* command. The short form of **clear capture** is not supported to prevent accidental destruction of all packet captures.

Note The **capture** command is not saved to the configuration, and the **capture** command is not replicated to the standby unit during failover.

Use the **copy capture:** *capture_name* **tftp://***location/path* **[pcap]** command to copy capture information to a remote TFTP server.

Use the **https://***pix-ip-address***/capture/***capture_name*[**/pcap**] command to view the packet capture information with a web browser.

If the **pcap** option is specified, then a libpcap-format file is downloaded to your web browser and can be saved using your web browser. (A libcap file can be viewed with Tcpdump or Ethereal.)

The **show capture** command displays the capture configuration when no options are specified. If the *capture_name* is specified, then it displays the capture buffer contents for that capture.

Output Formats

The decoded output of the packets are dependent on the protocol of the packet. In Table 4-3, the bracketed output is displayed when the **detail** option is specified.

Table 4-3 Packet Capture Output Formats

Packet Type	Capture Output Format
802.1Q	*HH:MM:SS.ms* [ether-hdr] *VLAN-info encap-ether-packet*
ARP	*HH:MM:SS.ms* [ether-hdr] *arp-type arp-info*
IP/ICMP	*HH:MM:SS.ms* [ether-hdr] *ip-source* > *ip-destination:* icmp: *icmp-type icmp-code* [checksum-failure]
IP/UDP	*HH:MM:SS.ms* [ether-hdr] *src-addr.src-port dest-addr.dst-port*: [checksum-info] udp *payload-len*
IP/TCP	*HH:MM:SS.ms* [ether-hdr] *src-addr.src-port* d*est-addr.dst-port*: *tcp-flags* [header-check] [checksum-info] *sequence-number ack-number tcp-window urgent-info tcp-options*
IP/Other	*HH:MM:SS.ms* [ether-hdr] *src-addr dest-addr*: *ip-protocol ip-length*
Other	*HH:MM:SS.ms ether-hdr*: *hex-dump*

Examples

On a web browser, the capture contents for a capture named “mycapture” can be viewed at the following location:

```
https://209.165.200.232/capture/mycapture/pcap
```

To download a libpcap file (used in web browsers such as Internet Explorer or Netscape Navigator) to a local machine, enter the following:

```
https://209.165.200.232/capture/http/pcap
```

In the following example, the traffic is captured from an outside host at 209.165.200.241 to an inside HTTP server.

```
access-list http permit tcp host 10.120.56.15 eq http host 209.165.200.241
access-list http permit tcp host 209.165.200.241 host 10.120.56.15 eq http
capture http access-list http packet-length 74 interface inside
```

To capture ARP packets, enter the following:

```
pixfirewall(config)# capture arp ethernet-type arp interface outside
```

To display the packets captured by an ARP capture, enter the following:

```
pixfirewall(config)# show capture arp
2 packets captured
19:12:23.478429 arp who-has 209.165.200.228 tell 209.165.200.10
19:12:26.784294 arp who-has 209.165.200.228 tell 209.165.200.10
2 packets shown
```

To capture PPPoE Discovery packets on multiple interfaces, enter the following:

```
pixfirewall(config)# capture pppoed ethernet-type pppoed interface outside
pixfirewall(config)# capture pppoed interface inside
```

The following stores a PPPoED trace to a file name "pppoed-dump" on a TFTP server at 209.165.201.17. (Some TFTP servers require that the file exists and is world writable, so check your TFTP server for the appropriate permissions and file first.)

```
pixfirewall(config)# copy capture:pppoed tftp://209.165.201.17/pppoed-dump
Writing to file '/tftpboot/pppoed-dump' at 209.165.201.17 on outside
```

To display the capture configuration, use the **show capture** command without specifying any options as follows:

```
pixfirewall(config)# show capture
capture arp ethernet-type arp interface outside
capture http access-list http packet-length 74 interface inside
```

clear

Removes configuration files and commands from the configuration, or resets command values. However, using the **no** form of a command is preferred to using the **clear** form to change your configuration because the **no** form is usually more precise.

clear file *configuration* | *pdm* | *pki*

clear *command*

no *command*

Command Modes

Configuration mode for **clear** commands that remove or reset firewall configurations. Privilege mode for commands that clear items such as counters in **show** commands. Additionally, the **clear** commands available in less secure modes are available in subsequent (more secure) modes. However, commands from a more secure mode are not available in a less secure mode.

Syntax Description

Table 4-4, Table 4-5, and Table 4-6 list the **clear** commands available in each mode.

Table 4-4 Unprivileged Mode Clear Command

Clear Command	Description	Used in the following command(s)
clear pager	Resets the number of displayed lines to 24.	**pager**

Table 4-5 Privileged Mode Clear Commands

Clear Command	Description	Used in the following command(s)
clear aaa accounting	To clear the local, TACACS+, or RADIUS user account.	**aaa accounting {include | exclude}**
clear aaa authentication	To clear the local or TACACS+ user authentication.	**aaa authentication**
clear aaa authorization	To clear the local or TACACS+ user authorization.	**aaa authorization {include | exclude}**
clear aaa-server	To remove a defined server group.	**aaa authorization,** **aaa authentication** **aaa-server**
clear arp	Clears the ARP table.	**arp**
clear auth-prompt	Removes an **auth-prompt** command statement from the configuration.	**auth-prompt**
clear banner	Removes all configured banners.	banner
clear blocks	Resets the **show blocks** command statement counters.	**show blocks/clear blocks**
clear configure	Resets command parameters in the configuration to their default values.	**configure**
clear crashinfo	Deletes the crash information file from the Flash memory of the firewall.	crashinfo
clear flashfs	Clears Flash memory prior to downgrading the PIX Firewall software version.	**fragment**
clear floodguard	Removes Flood Defender, which protects against flood attacks from configuration.	**floodguard**
clear local-host	Resets the information displayed for the **show local-host** command.	**show local-host/clear local host**
clear passwd	Resets the Telnet password back to “cisco.”	**password**
clear traffic	Resets the counters for the **show traffic** command.	**show traffic/clear traffic**
clear uauth	Deletes one user’s or all users’ AAA authorization caches, which forces the users to reauthenticate the next time they create a connection.	**show uauth/clear uauth**
clear xlate	Clears the contents of the translation slots.	**show xlate/clear xlate**

Table 4-6 Configuration Mode Clear Commands

Clear Command	Description	Used in the following command(s)
clear aaa	Removes **aaa** command statements from the configuration.	**aaa accounting**
clear aaa accounting	Removes **aaa-server** command statements from the configuration.	aaa authorization
clear aaa-server	Remove a defined server group from the configuration.	aaa authorization
clear access-group	Removes **access-group** command statements from the configuration.	**access-group**
clear access-list	Removes **access-list** command statements from the configuration. This command also stops all traffic through the PIX Firewall on the affected **access-list** command statements.	**access-list**
clear access-list *aclname* **counters**	Clears the counters shown by the **show access-list** command.	**access-list**
clear alias	Removes **alias** command statements from the configuration.	**alias**
clear apply	Removes **apply** command statements from the configuration.	**outbound/apply**
clear capture	Clears the packet capture.	capture
clear clock	Removes **clock** command statements from the configuration.	clock
clear conduit	Removes **conduit** command statements from the configuration.	**conduit**
clear dhcpd	Removes **dhcpd** command statements from the configuration.	**dhcpd**
clear established	Removes **established** command statements from the configuration.	**established**
clear filter	Removes **filter** command statements from the configuration.	**filter**
clear fixup	Resets **fixup protocol** command statements to their default values.	**fixup protocol**
clear flashfs	Clears Flash memory before downgrading to a previous PIX Firewall version.	**fragment**
clear global	Removes **global** command statements from the configuration.	**global**
clear http	Removes all HTTP hosts and disables the server.	http
clear icmp	Removes **icmp** command statements from the configuration.	**icmp**
clear ip	Sets all PIX Firewall interface IP addresses to 127.0.0.1 and stops all traffic.	**ip address**

Table 4-6 Configuration Mode Clear Commands (continued)

Clear Command	Description	Used in the following command(s)
clear ip address	Clears all PIX Firewall interface IP addresses (configuration mode).	**ip address**
clear ip audit	Clears the IDS signature on the interface (configuration mode).	**ip audit**
clear ip local pool	Clears pool of local IP addresses for dynamic assignment to a VPN.	ip local pool
clear ip verify reverse-path	Clears RPF IP spoofing protection (configuration mode).	**ip verify reverse-path**
clear [**crypto**] **dynamic-map**	Remove **crypto dynamic-map** command statements from the configuration.The keyword **crypto** is optional.	**crypto dynamic-map** and **dynamic-map**
clear [**crypto**] **ipsec sa**	Delete the active IPSec security associations. The keyword **crypto** is optional.	**crypto ipsec**
clear [**crypto**] **ipsec sa counters**	Clear the traffic counters maintained for each security association. The keyword **crypto** is optional.	**crypto ipsec**
clear [**crypto**] **ipsec sa entry** *destination-address protocol spi*	Delete the active IPSec security association with the specified address, protocol, and SPI. The keyword **crypto** is optional.	**crypto ipsec**
clear [**crypto**] **ipsec sa map** *map-name*	Delete the active IPSec security associations for the named crypto map set. The keyword **crypto** is optional.	**crypto ipsec**
clear [**crypto**] **ipsec sa peer**	Delete the active IPSec security associations for the specified peer. The keyword **crypto** is optional.	**crypto ipsec**
clear [**crypto**] **isakmp sa**	Delete the active IKE security associations. The keyword **crypto** is optional.	**isakmp**
clear [crypto] map	Delete all parameters entered through the **crypto map** command belonging to the specified map. Does not delete dynamic maps.	crypto map
clear isakmp	Remove **isakmp** command statements from the configuration.	**isakmp**
clear isakmp log	Clears events in the isakmp log buffer	**isakmp**
clear interface	Clear counters for the **show interface** command.	**interface**
clear logging	Clear syslog message queue accumulated by the **logging buffered** command.	**logging**
clear names	Removes **name** command statements from the configuration.	**name/names**
clear nameif	Reverts **nameif** command statements to default interface names and security levels.	**nameif**

Table 4-6 Configuration Mode Clear Commands (continued)

Clear Command	Description	Used in the following command(s)
clear nat	Removes **nat** command statements from the configuration.	**nat**
clear ntp	Removes **ntp** command statements from the configuration.	ntp
clear outbound	Removes **outbound** command statements from the configuration.	**outbound/apply**
clear ospf [*process-id*] {**process** | **counters** | **neighbor** [**neighbor-intf**] [**neighbr-id**]}	Clears and restarts the OSPF process with the specified ID, resets OSPF interface counters, neighbor interface router designation, or neighbor router ID, depending on the option selected. This command does not remove any configuration. Use the no form of the **router ospf** or **routing interface** command to remove the OSPF configuration.	routing interface
clear pdm	Removes all locations, disables logging and clears the PDM buffer. Internal PDM command.	pdm
clear privilege	Removes **privilege** command statements from the configuration.	privilege
clear rip	Removes **rip** command statements from the configuration.	**rip**
clear route	Removes **route** command statements from the configuration that do not contain the CONNECT keyword.	**route**
clear service	Removes **service** command statements from the configuration.	**service**
clear snmp-server	Removes **snmp-server** command statements from the configuration.	**· When this feature is off, regular SIP Fixup will work as it does under PIX 6.3.3**
clear ssh	Removes **ssh** command statement from the configuration.	**ssh**
clear static	Removes **static** command statements from the configuration.	**static**
clear sysopt	Removes **sysopt** command statements from the configuration.	**sysopt**
clear telnet	Removes **telnet** command statements from the configuration.	**telnet**
clear tftp-server	Removes **tftp-server** command statements from the configuration.	**tftp-server**
clear timeout	Resets **timeout** command durations to their default values.	**timeout**
clear url-cache	Removes **url-cache** command statements from the configuration.	**url-cache**

Table 4-6 Configuration Mode Clear Commands (continued)

Clear Command	Description	Used in the following command(s)
clear url-server	Removes **url-server** command statements from the configuration.	**url-server**
clear username	Removes **username** command statements from the configuration.	username
clear virtual	Removes **virtual** command statements from the configuration.	virtual
clear vpdn	Removes **vpdn** command statements from the configuration.	vpdn
clear vpnclient	Removes **vpnclient** command statements from the configuration.	vpnclient

clock

Set the PIX Firewall clock for use with the PIX Firewall Syslog Server (PFSS) and the Public Key Infrastructure (PKI) protocol.

clock set *hh*:*mm*:*ss* {*day month* | *month day*} *year*

clear clock

*[***no***]* **clock summer-time** *zone* **recurring** [*week weekday month hh*:*mm week weekday month hh*:*mm] [offset]*

*[***no***]* **clock summer-time** *zone* **date** {*day month* | *month day*} *year hh*:*mm* {*day month* | *month day*} *year hh*:*mm [offset]*

*[***no***]* **clock timezone** *zone hours [minutes]*

show clock [detail]

Syntax Description

date	The **date** command form is used as an alternative to the **recurring** form of the **clock summer-time** command. It specifies that summertime should start on the first date entered and end on the second date entered. If the start date month is after the end date month, the summer time zone is accepted and assumed to be in the Southern Hemisphere.
day	The day of the month to start, from 1 to 31.
detail	Displays the clock source and current summertime settings.
hh:*mm*:*ss*	The hour:minutes:seconds expressed in 24-hour time; for example, **20:54:00** for 8:54 pm. Zeros can be entered as a single digit; for example, **21:0:0**.
hours	The hours of offset from UTC.
minutes	The minutes of offset from UTC.
month	The month expressed as the first three characters of the month; for example, **apr** for April.
offset	The number of minutes to add during summertime. The default is 60 minutes.
recurring	Specifies the start and end dates for local summer "daylight savings" time. The first date entered is the start date and the second date entered is the end date. (The start date is relative to UTC and the end date is relative to the specified summer time zone.) If no dates are specified, United States Daylight Savings Time is used. If the start date month is after the end date month, the summer time zone is accepted and assumed to be in the Southern Hemisphere.
summer-time	The **clock summer-time** command displays summertime hours during the specified summertime date range. This command affects the clock display time only.
timezone	**clock timezone** sets the clock display to the time zone specified. It does not change internal PIX Firewall time, which remains UTC.
week	Specifies the week of the month. The week is 1 through 4 and **first** or **last** for partial weeks at the begin or end a month, respectively. For example, week 5 of any month is specified by using **last**.
weekday	Specifies the day of the week: Monday, Tuesday, Wednesday, and so on.

year	The year expressed as four digits; for example, **2000**. The year range supported for the **clock** command is 1993 to 2035.
zone	The name of the time zone.

Command Modes

Configuration mode.

Usage Guidelines

The **clock** command lets you specify the time, month, day, and year for use with time stamped syslog messages, which you can enable with the **logging timestamp** command. You can view the time with the **clock** or the **show clock** command.

The **clear clock** command removes all summertime settings and resets the clock display to UTC.

The **show clock** command outputs the time, time zone, day, and full date.

Note The lifetime of a certificate and the certificate revocation list (CRL) is checked in UTC, which is the same as GMT. If you are using IPSec with certificates, set the PIX Firewall clock to UTC to ensure that CRL checking works correctly.

You can interchange the settings for the *day* and the *month*; for example, **clock set 21:0:0 1 apr 2000**.

The maximum date range for the **clock** command is 1993 through 2035. A time prior to January 1, 1993, or after December 31, 2035, will not be accepted.

While the PIX Firewall clock is year 2000 compliant, it does not adjust itself for daylight savings time changes; however, it does know about leap years.

The PIX Firewall clock setting is retained in memory when the power is off by a battery on the PIX Firewall unit's motherboard. Should this battery fail, contact Cisco TAC for a replacement PIX Firewall unit.

Cisco's PKI (Public Key Infrastructure) protocol uses the clock to make sure that a certificate revocation list (CRL) is not expired. Otherwise, the CA may reject or allow certificates based on an incorrect timestamp. Refer to the *Cisco PIX Firewall and VPN Configuration Guide* for a description of IPSec concepts.

Examples

To enable PFSS time stamp logging for the first time, use the following commands:

```
clock set 21:0:0 apr 1 2000
show clock
21:00:05 Apr 01 2000
logging host 209.165.201.3
logging timestamp
logging trap 5
```

In this example, the **clock** command sets the clock to 9 p.m. on April 1, 2000. The **logging host** command specifies that a syslog server is at IP address 209.165.201.3. The PIX Firewall automatically determines that the server is a PFSS and sends syslog messages to it via TCP and UDP. The **logging timestamp** command enables sending time stamped syslog messages. The **logging trap 5** command in this example specifies that messages at syslog level 0 through 5 be sent to the syslog server. The value 5 is used to capture severe and normal messages, but also those of the **aaa authentication enable** command.

The following **clock summer-time** command specifies that summertime starts on the first Sunday in April at 2 a.m. and ends on the last Sunday in October at 2 a.m.:

```
pix_name (config)# clock summer-time PDT recurring 1 Sunday April 2:00
last Sunday October 2:00
```

If you live in a place where summertime follows the Southern Hemisphere pattern, you can specify the exact date and times. In the following example, daylight savings time (summer time) is configured to start on October 12, 2001, at 2 a.m. and end on April 26, 2002, at 2 a.m.:

```
pix_name (config)# clock summer-time PDT date 12 October 2001 2:00
26 April 2002 2:00
```

conduit

Add, delete, or show conduits through the PIX Firewall for incoming connections. However, the **conduit** command has been superseded by the **access-list** command. We recommend that you migrate your configuration away from the **conduit** command to maintain future compatibility.

*[***no***]* **conduit permit | deny** *protocol global_ip global_mask* [*operator port* [*port*]] *foreign_ip foreign_mask* [*operator port* [*port*]]

*[***no***]* **conduit deny|permit** *protocol* **| object-group** *protocol_obj_grp_id global_ip global_mask* **| object-group** *network_obj_grp_id* **[***operator port* **[***port***] | object-group** *service_obj_grp_id***]** *foreign_ip foreign_mask* **| object-group** *network_obj_grp_id* **[***operator port* **[***port***] | object-group** *service_obj_grp_id***]**

*[***no***]* **conduit deny|permit icmp** *global_ip global_mask* **| object-group** *network_obj_grp_id foreign_ip foreign_mask* **| object-group** *network_obj_grp_id* [*icmp_type* **| object-group** *icmp_type_obj_grp_id***]**

clear conduit

clear conduit counters

show conduit

Syntax Description

deny	Deny access if the conditions are matched.
foreign_ip	An external IP address (host or network) that can access the *global_ip*. You can specify **0.0.0.0** or **0** for any host. If both the *foreign_ip* and *foreign_mask* are 0.0.0.0 0.0.0.0, you can use the shorthand **any** option. If *foreign_ip* is a host, you can omit *foreign_mask* by specifying the **host** command before *foreign_ip*. For example: **conduit permit tcp any eq ftp host 209.165.201.2** This example lets foreign host 209.165.201.2 access any global address for FTP.

foreign_mask	Network mask of *foreign_ip*. The *foreign_mask* is a 32-bit, four-part dotted decimal; such as, 255.255.255.255. Use zeros in a part to indicate bit positions to be ignored. Use subnetting if required. If you use **0** for *foreign_ip*, use **0** for the *foreign_mask*; otherwise, enter the *foreign_mask* appropriate to *foreign_ip*. You can also specify a mask for subnetting. For example: 255.255.255.192.
global_ip	A global IP address previously defined by a **global** or **static** command. You can use **any** if the *global_ip* and *global_mask* are 0.0.0.0 0.0.0.0. The **any** option applies the **permit** or **deny** parameters to the global addresses. If *global_ip* is a host, you can omit *global_mask* by specifying the **host** command before *global_ip*. For example: `conduit permit tcp host 209.165.201.1 eq ftp any` This example lets any foreign host access global address 209.165.201.1 for FTP.
global_mask	Network mask of *global_ip*. The *global_mask* is a 32-bit, four-part dotted decimal; such as, 255.255.255.255. Use zeros in a part to indicate bit positions to be ignored. Use subnetting if required. If you use **0** for *global_ip*, use **0** for the *global_mask*; otherwise, enter the *global_mask* appropriate to *global_ip*.
icmp_type	The type of ICMP message. Table 4-7 lists the ICMP type literals that you can use in this command. Omit this option to include all ICMP types. The **conduit permit icmp any any** command permits all ICMP types and lets ICMP pass inbound and outbound.
icmp_type _obj_grp_id	An existing ICMP type object group.
object-group	Specifies an object group.

operator	A comparison operand that lets you specify a port or a port range.
	
	Use without an operator and port to indicate all ports.
	
	For example:
	
	`conduit permit tcp any any`
	
	Use **eq** and a port to permit or deny access to just that port. For example use **eq ftp** to permit or deny access only to FTP:
	
	`conduit deny tcp host 209.165.200.247 eq ftp 209.165.201.1`
	
	Use **lt** and a port to permit or deny access to all ports less than the port you specify. For example, use **lt 2025** to permit or deny access to the well-known ports (1 to 1024).
	
	`conduit permit tcp host 209.165.200.247 lt 1025 any`
	
	Use **gt** and a port to permit or deny access to all ports greater than the port you specify. For example, use **gt 42** to permit or deny ports 43 to 65535.
	
	`conduit deny udp host 209.165.200.247 gt 42 host 209.165.201.2`
	
	Use **neq** and a port to permit or deny access to every port except the ports that you specify. For example, use **neq 10** to permit or deny ports 1-9 and 11 to 65535.
	
	`conduit deny tcp host 209.165.200.247 neq 10 host 209.165.201.2 neq 42`
	
	Use **range** and a port range to permit or deny access to only those ports named in the range. For example, use **range 10 1024** to permit or deny access only to ports 10 through 1024. All other ports are unaffected.
	
	`conduit deny tcp any range ftp telnet any`
	
	By default, all ports are denied until explicitly permitted.

network_obj_grp_id	An existing network object group.
permit	Permit access if the conditions are matched.
port	Service(s) you permit to be used while accessing *global_ip or foreign_ip*. Specify services by the port that handles it, such as **smtp for port 25**, **www** for port 80, and so on. You can specify ports by either a literal name or a number in the range of 0 to 65535. You can specify all ports by not specifying a port value. For example: `conduit deny tcp any any` This command is the default condition for the **conduit** command in that all ports are denied until explicitly permitted. You can view valid port numbers online at the following website: http://www.iana.org/assignments/port-numbers See ""Ports""in Chapter 2, "Using PIX Firewall Commands" for a list of valid port literal names in port ranges; for example, **ftp h323**. You can also specify numbers.

protocol	Specify the transport protocol for the connection. Possible literal values are **icmp**, **tcp**, **udp**, or an integer in the range 0 through 255 representing an IP protocol number. Use **ip** to specify all transport protocols. You can view valid protocol numbers online at the following website: http://www.iana.org/assignments/protocol-numbers If you specify the **icmp** protocol, you can permit or deny ICMP access to one or more global IP addresses. Specify the ICMP type in the *icmp_type* variable, or omit to specify all ICMP types. See "Usage Guidelines" for a complete list of the ICMP types.
protocol_obj_grp_id	An existing protocol object group.
service_obj_grp_id	An existing service (port) object group.

Command Modes

Configuration mode.

Usage Guidelines

We recommend that you use the **access-list** command instead of the **conduit** command because using an access list is a more secure way of enabling connections between hosts. Specifically, the **conduit** command functions by creating an exception to the PIX Firewall Adaptive Security Algorithm that then permits connections from one PIX Firewall network interface to access hosts on another.

The **conduit** command can permit or deny access to either the **global** or **static** commands; however, neither is required for the **conduit** command. You can associate a **conduit** command statement with a **global** or **static** command statement through the global address, either specifically to a single global address, a range of global addresses, or to all global addresses.

When used with a **static** command statement, a **conduit** command statement permits users on a lower security interface to access a higher security interface. When not used with a **static** command statement, a **conduit** command statement permits both inbound and outbound access.

The **show conduit** command displays the **conduit** command statements in the configuration and the number of times (hit count) an element has been matched during a **conduit** command search.

Converting conduit Commands to access-list Commands

Follow these steps to convert **conduit** command statements to **access-list** commands:

Step 1 View the **static** command format. This command normally precedes both the **conduit** and **access-list** commands. The **static** command syntax is as follows.

static (*high_interface*,*low_interface*) *global_ip local_ip* **netmask** *mask*

For example:

```
static (inside,outside) 209.165.201.5 192.168.1.5 netmask 255.255.255.255
```

This command maps the global IP address 209.165.201.5 on the outside interface to the web server 192.168.1.5 on the inside interface. The 255.255.255.255 is used for host addresses.

Step 2 View the **conduit** command format. The **conduit** command is similar to the **access-list** command in that it restricts access to the mapping provided by the **static** command. The **conduit** command syntax is as follows.

conduit *action protocol global_ip global_mask global_operator global_port* [*global_port*] *foreign_ip foreign_mask foreign_operator foreign_por*t [*foreign_port*]

For example:

```
conduit permit tcp host 209.165.201.5 eq www any
```

This command permits TCP for the global IP address 209.165.201.5 that was specified in the **static** command statement and permits access over port 80 (**www**). The "**any**" option lets any host on the outside interface access the global IP address.

The **static** command identifies the interface that the **conduit** command restricts access to.

Step 3 Create the **access-list** command from the **conduit** command options. The *acl_name* in the **access-list** command is a name or number you create to associate **access-list** command statements with an **access-group** or **crypto map** command statement.

Normally the **access-list** command format is as follows:

access-list *acl_name* [**deny** | **permit**] *protocol src_addr src_mask operator port dest_addr dest_mask operator port*

However, using the syntax from the **conduit** command in the **access-list** command, you can see how the *foreign_ip* in the **conduit** command is the same as the *src_addr* in the **access-list** command and how the *global_ip* option in the **conduit** command is the same as the *dest_addr* in the **access-list** command. The **access-list** command syntax overlaid with the **conduit** command options is as follows.

access-list *acl_name action protocol foreign_ip foreign_mask foreign_operator foreign_por*t [*foreign_port*] *global_ip global_mask global_operator global_port* [*global_port*]

For example:

```
access-list acl_out permit tcp any host 209.165.201.5 eq www
```

This command identifies the **access-list** command statement group with the "**acl_out**" identifier. You can use any name or number for your own identifier. (In this example the identifier, "acl" is from ACL, which means access control list and "out" is an abbreviation for the outside interface.) It makes your configuration clearer if you use an identifier name that indicates the interface to which you are associating the **access-list** command statements. The example **access-list** command, like the **conduit** command, permits TCP connections from any system on the outside interface. The **access-list** command is associated with the outside interface with the **access-group** command.

Step 4 Create the **access-group** command using the *acl_name* from the **access-list** command and the *low_interface* option from the **static** command. The format for the **access-group** command is as follows.

access-group *acl_name* **in interface** *low_interface*

For example:

```
access-group acl_out in interface outside
```

This command associates with the "**acl_out**" group of **access-list** command statements and states that the **access-list** command statement restricts access to the outside interface.

More on the conduit Command

If you associate a **conduit** command statement with a **static** command statement, only the interfaces specified on the **static** command statement have access to the **conduit** command statement. For example, if a **static** command statement lets users on the **dmz** interface access a server on the inside interface, only users on the **dmz** interface can access the server via the **static** command statement. Users on the outside do not have access.

Note The **conduit** command statements are processed in the order they are entered into the configuration.

The **permit** and **deny** options for the **conduit** command are processed in the order listed in the PIX Firewall configuration. In the following example, host 209.165.202.129 is not denied access through the PIX Firewall because the **permit** option precedes the **deny** option.

```
conduit permit tcp host 209.165.201.4 eq 80 any
conduit deny tcp host 209.165.201.4 host 209.165.202.129 eq 80 any
```

Note If you want internal users to be able to ping external hosts, use the **conduit permit icmp any any** command.

After changing or removing a **conduit** command statement, use the **clear xlate** command.

You can remove a **conduit** command statement with the **no conduit** command. The **clear conduit** command removes all **conduit** command statements from your configuration. The **clear conduit counters** command clears the current conduit hit count.

If you prefer more selective ICMP access, you can specify a single ICMP message type as the last option in this command. Table 4-7 lists possible ICMP types values.

Table 4-7 ICMP Type Literals

ICMP Type	Literal
0	echo-reply
3	unreachable
4	source-quench
5	redirect
6	alternate-address
8	echo
9	router-advertisement
10	router-solicitation
11	time-exceeded
12	parameter-problem
13	timestamp-request
14	timestamp-reply
15	information-request
16	information-reply
17	mask-request
18	mask-reply

Table 4-7 ICMP Type Literals (continued)

ICMP Type	Literal
31	conversion-error
32	mobile-redirect

Usage Notes

1. By default, all ports are denied until explicitly permitted.
2. The **conduit** command statements are processed in the order entered in the configuration. If you remove a command, it affects the order of all subsequent **conduit** command statements.
3. To remove all **conduit** command statements, cut and paste your configuration onto your console computer, edit the configuration on the computer, use the **write erase** command to clear the current configuration, and then paste the configuration back into the PIX Firewall.
4. If you use Port Address Translation (PAT), you cannot use a **conduit** command statement using the PAT address to either permit or deny access to ports.
5. Two **conduit** command statements are required for establishing access to the following services: **discard**, **dns**, **echo**, **ident**, **pptp**, **rpc**, **sunrpc**, **syslog**, **tacacs-ds**, **talk**, and **time**. Each service, except for **pptp**, requires one **conduit** for TCP and one for UDP. For DNS, if you are only receiving zone updates, you only need a single **conduit** command statement for TCP.

   The two **conduit** command statements for the PPTP transport protocol, which is a subset of the GRE protocol, are as shown in the following example:

   ```
   static (dmz2,outside) 209.165.201.5 192.168.1.5 netmask 255.255.255.255
   conduit permit tcp host 209.165.201.5 eq 1723 any
   conduit permit gre host 209.165.201.5 any
   ```

   In this example, PPTP is being used to handle access to host 192.168.1.5 on the **dmz2** interface from users on the outside. Outside users access the dmz2 host using global address 209.165.201.5. The first **conduit** command statement opens access for the PPTP protocol and gives access to any outside users. The second **conduit** command statement permits access to GRE. If PPTP was not involved and GRE was, you could omit the first **conduit** command statement.
6. The RPC **conduit** command support fixes up UDP portmapper and rpcbind exchanges. TCP exchanges are not supported. This lets simple RPC-based programs work; however, remote procedure calls, arguments, or responses that contain addresses or ports will not be fixed up.

   For MSRPC, two **conduit** command statements are required, one for port 135 and another for access to the high ports (1024-65535). For Sun RPC, a single **conduit** command statement is required for UDP port 111.

   Once you create a **conduit** command statement for RPC, you can use the following command to test its activity from a UNIX host:

   ```
   rpcinfo -u unix_host_ip_address 150001
   ```

   Replace *unix_host_ip_address* with the IP address of the UNIX host.
7. You can overlay host statics on top of a net static range to further refine what an individual host can access:

   ```
   static (inside, outside) 209.165.201.0 10.1.1.0 netmask 255.255.255.0
   conduit permit tcp 209.165.201.0 255.255.255.0 eq ftp any
   static (inside, outside) 203.31.17.3 10.1.1.3 netmask 255.255.255.0
   conduit permit udp host 209.165.201.3 eq h323 host 209.165.202.3
   ```

In this case, the host at 209.165.202.3 has Intel Internet Phone access in addition to its blanket FTP access.

Examples

1. The following commands permit access between an outside UNIX gateway host at 209.165.201.2, to an inside SMTP server with Mail Guard at 192.168.1.49. Mail Guard is enabled in the default configuration for PIX Firewall with the **fixup protocol smtp 25** command. The global address on the PIX Firewall is 209.165.201.1.

```
static (inside,outside) 209.165.201.1 192.168.1.49 netmask 255.255.255.255 0 0
conduit permit tcp host 209.165.201.1 eq smtp host 209.165.201.2
```

   To disable Mail Guard, enter the following command:

```
no fixup protocol smtp 25
```

2. You can set up an inside host to receive H.323 Intel Internet Phone calls and allow the outside network to connect inbound via the IDENT protocol (TCP port 113). In this example, the inside network is at 192.168.1.0, the global addresses on the outside network are referenced via the 209.165.201.0 network address with a 255.255.255.224 mask.

```
static (inside,outside) 209.165.201.0 192.168.1.0 netmask 255.255.255.224 0 0
conduit permit tcp 209.165.201.0 255.255.255.224 eq h323 any
conduit permit tcp 209.165.201.0 255.255.255.224 eq 113 any
```

3. You can create a web server on the perimeter interface that can be accessed by any outside host as follows:

```
static (perimeter,outside) 209.165.201.4 192.168.1.4 netmask 255.255.255.255 0 0
conduit permit tcp host 209.165.201.4 eq 80 any
```

   In this example, the **static** command statement maps the perimeter host, 192.168.1.4. to the global address, 209.165.201.4. The **conduit** command statement specifies that the global host can be accessed on port 80 (web server) by any outside host.

configure

Configure from the terminal, Flash memory, the network, or factory default. The new configuration merges with the active configuration except for the factory default, in which case the active configuration is cleared first and then replaced by the factory default. The factory default option is available only on the PIX 501 and PIX 506/506E.

clear configure [**terminal | memory**]

clear configure [**primary | secondary | all**]

[**no**] **configure http**[**s**] **://** [*user:password@*] *location* [**:***port*] **/** *http_pathname*

configure net [[*location*]**:**[*filename*]]

clear configure primary | secondary | all

show configure

For the PIX 501 and PIX 506/506E only:
configure factory-default [*inside_ip_address* [*address_mask*]]

For older PIX Firewall units that have a floppy drive only:
configure floppy

Syntax Description

address_mask	Specifies the address mask for the inside interface IP address. The default address mask is 255.255.255.0.
all	Combines the **primary** and **secondary** options.
clear	Clears aspects of the current configuration in RAM. Use the **write erase** command to clear the complete configuration.
factory-default	Specifies to clear the current configuration and regenerate the default, factory-loaded configuration. This command is supported for the PIX 501 and PIX 506/506E only in PIX Firewall software Version 6.2.
filename	A filename you specify to qualify the location of the configuration file on the TFTP server named in *server_ip*. If you set a filename with the **tftp-server** command, do not specify it in the **configure** command; instead just use a colon (**:**) without a filename.
floppy	Merges the current configuration with that on diskette.
http_pathname	The name of the HTTP server path that contains the PIX Firewall configuration to copy.
http[**s**]	Specifies to retrieve configuration information from an HTTP server. (SSL is used when **https** is specified.)
inside_ip_address	Specifies the inside IP address. The default inside interface IP address is 192.168.1.1.
location	The IP address (or defined name) of the HTTP server to log into.
memory	Merges the current configuration with that in Flash memory.
net	Loads the configuration from a TFTP server and the path you specify. Comments in the configuration preceded by a colon (:) or exclamation mark (!) will be pruned and will not be visible in the PIX Firewall configuration listing.
password	The password for logging into the HTTP server.
pathname	The name of the resource that contains the PIX Firewall configuration to copy.
port	Specifies the port to contact on the HTTP server. It defaults to 80 for **http** and 443 for **https**.
primary	Sets the **interface**, **ip**, **mtu**, **nameif**, and **route** commands to their default values. In addition, interface names are removed from all commands in the configuration.
secondary	Removes the **aaa-server**, **alias**, **access-list**, **apply**, **conduit**, **global**, **outbound**, **static**, **telnet**, and **url-server** command statements from your configuration.
location	The IP address or name of the server from which to merge in a new configuration. This server address or name is defined with the **tftp-server** command.
terminal	Starts configuration mode to enter configuration commands from a terminal. Exit configuration mode by entering the **quit** command.
user	The username for logging into the HTTP server.

Command Modes

The **configure terminal** command (with the short form "**config t**") is available in privileged mode, and it changes the firewall over to configuration mode. All other **configure** commands are available in configuration mode.

Usage Guidelines

You must be in configuration mode to use the **configuration** commands, except for the **configure terminal** (**config t**) command. The **configure terminal** command starts configuration mode from privileged mode. You can exit configuration mode with the **quit** command. After exiting configuration mode, use the **write memory** command to store your changes in Flash memory or **write floppy** to store the configuration on diskette.

Each command statement from Flash memory (with **configure memory**), TFTP transfer (with **configure net**), or diskette (with **configure floppy**) is read into the current configuration and evaluated in the same way as commands entered from a keyboard with the following rules:

- If the command in Flash memory or on diskette is identical to an existing command in the current configuration, it is ignored.
- If the command in Flash memory or on diskette is an additional instance of an existing command, such as if you already have one **telnet** command for IP address 10.2.3.4 and the diskette configuration has a **telnet** command for 10.7.8.9, then both commands appear in the current configuration.
- If the command redefines an existing command, the command on diskette or Flash memory overwrites the command in the current configuration in RAM. For example, if you have the **hostname ram** command in the current configuration and the **hostname floppy** command on diskette, the command in the configuration becomes **hostname floppy** and the command line prompt changes to match the new hostname when that command is read from diskette.

The **show configure** and **show startup-config** commands display the startup configuration of the firewall. The **write terminal** and **show running-config** commands display the configuration currently running on the firewall.

The **clear configure** [**all**] command resets a configuration to its default values. Use this command to create a template configuration or when you want to clear all values. The **clear configure primary** command resets the default values for the **interface**, **ip**, **mtu**, **nameif**, and **route** commands. This command also deletes interface names in the configuration. The **clear configure secondary** command removes the **aaa-server**, **alias**, **access-list**, **apply**, **conduit**, **global**, **outbound**, **static**, **telnet**, and **url-server** command statements from the configuration. However, the **clear configure secondary** command does not remove **tftp-server** command statements.

Note Save your configuration before using a **clear configure** command. The **clear configure primary** and **clear configure secondary** commands do not prompt you before deleting lines from your configuration.

configure factory-default

On the PIX 501 and PIX 506/506E, the **configure factory-default** command reinstates the factory default configuration. (This command is not supported on other PIX Firewall platforms at this time.) Use this command carefully because, before reinstating the factory default configuration, this command has the same effect as the **clear configure all** command; it clears all existing configuration information.

With no options specified, the **configure factory-default** command gives a default IP address of 192.168.1.1, and a netmask of 255.255.255.0, to the PIX Firewall inside interface.

With the **configure factory-default ip-address** command, if you specify an inside IP address but no netmask, the default address mask is derived from the specified IP address and is based on the IP address class.

With the **configure factory-default ip-address netmask** command, the specified IP address and netmask are assigned to the inside interface of the firewall.

For the PIX 501, the 10-user license is limited to a DHCP pool of 32 addresses, the 50-user license is limited to a DHCP pool size of 128 addresses, and the unlimited user license is limited to a DHCP pool size of 253 addresses. (It would be 256 addresses for the unlimited user license, but the default IP address is class C and 256 DHCP addresses cannot be supported within a class C address.) The PIX 506/506E is limited to a DHCP pool size of 253.

configure http[s]

The **configure http**[**s**] command retrieves configuration information from an HTTP server for remotely managing a PIX Firewall configuration. The configuration can be either a text file or an XML file. Text files merge regardless of errors that may be in the cofiguration. XML files require the use of the message "config-data" in the XML file to explicitly control merging and error handling.

configure net

The **configure net** command merges the current running configuration with a TFTP configuration stored at the IP address you specify and from the file you name. If you specify both the IP address and path name in the **tftp-server** command, you can specify *server_ip*:*filename* as simply a colon (**:**).

For example:

```
configure net :
```

Use the **write net** command to store the configuration in the file.

If you have an existing PIX Firewall configuration on a TFTP server and store a shorter configuration with the same filename on the TFTP server, some TFTP servers will leave some of the original configuration after the first ":end" mark. This does not affect the PIX Firewall because the **configure net** command stops reading when it reaches the first ":end" mark. However, this may cause confusion if you view the configuration and see extra text at the end of the configuration.

Note Many TFTP servers require the configuration file to be world-readable to be accessible.

configure floppy

The **configure floppy** command merges the current running configuration with the configuration stored on diskette. This command assumes that the diskette was previously created by the **write floppy** command.

configure memory

The **configure memory** command merges the configuration in Flash memory into the current configuration in RAM.

Examples

The following example shows how to configure the PIX Firewall using a configuration retrieved with TFTP:

```
configure net 10.1.1.1:/tftp/config/pixconfig
```

The pixconfig file is stored on the TFTP server at 10.1.1.1 in the tftp/config folder.

The following example shows how to configure the PIX Firewall from a diskette:

```
configure floppy
```

The following example shows how to configure the PIX Firewall from the configuration stored in Flash memory:

```
configure memory
```

The following example shows the commands you enter to access configuration mode, view the configuration, and save it in Flash memory.

Access privileged mode with the **enable** command and configuration mode with the **configure terminal** command. View the current configuration with the **write terminal** command and save your configuration to Flash memory using the **write memory** command.

```
pixfirewall> enable
password:
pixfirewall# configure terminal
pixfirewall(config)# write terminal
:  Saved
[...current configuration...]
:  End

write memory
```

When you enter the **configure factory-default** command on a platform other than the PIX 501 or PIX 506/506E, the PIX Firewall displays a "not supported" error message. On the PIX 515/515E, for example, the following message is displayed:

```
pixdfirewall(config)# configure factory default
'config factory-default' is not supported on PIX-515
```

console

Sets the idle timeout for the serial-cable console session of the PIX Firewall.

[**no**] **console timeout** *number*

Syntax Description

number	Idle time in minutes (0-60) after which the serial-cable console session ends.

Defaults

The default timeout is 0, which means the console will not time out. The zero value in the command **console timeout 0** has the same meaning as zero value in the command **exec-timeout 0 0** in Cisco IOS software.

Command Modes

The **console timeout** command is available in configuration mode.

The **show console timeout** command is available in privileged and configuration mode.

Usage Guidelines

The **console timeout** command sets the timeout value for any authenticated, enable mode, or configuration mode user session when accessing the firewall console through a serial cable. This timeout does not alter the Telnet or SSH timeouts; these access methods maintain their own timeout values.

The **no console timeout** command resets the console timeout value to its default.

The **show console timeout** command displays the currently configured console timeout value.

Examples

The following example shows how to set the console timeout to fifteen (15) minutes:

```
pixfirewall(config)# console timeout 15
```

The following example shows how to display the configured timeout value:

```
pixfirewall(config)# show console timeout
console timeout 15
```

Related Commands

aaa authorization	Enable or disable LOCAL or TACACS+ user authorization services.
password	Sets the password for Telnet access to the PIX Firewall console.
ssh	Specifies a host for PIX Firewall console access through Secure Shell (SSH).
telnet	Specifies the host for PIX Firewall console access via Telnet.

copy

Change software images without requiring access to the TFTP monitor mode or copy a capture file to a TFTP server.

copy capture: *capture_name* **tftp://***location/path* [**pcap**]

copy http[**s**]**://**[*user:password@*] *location* [*:port*] **/** *http_pathname* **flash** [**:** [**image** | **pdm**]]

copy tftp[**:**[[**//***location*] [**/***tftp_pathname*]]] **flash**[**:**[**image** | **pdm**]]

Syntax Description

copy capture *capture_name*	Copies capture information to a remote TFTP server. *capture_name* is a unique name that identifies the capture.
copy http[**s**]	Downloads a software image into the Flash memory of the firewall from an HTTP server. (SSL is used when **https** is specified.)
copy tftp flash	Downloads a software image into Flash memory of the firewall via TFTP without using monitor mode.
http_pathname	The name of the resource that contains the PIX Firewall software image or PDM file to copy.
image	Download the selected PIX Firewall image to Flash memory. An image you download is made available to the PIX Firewall on the next reload (reboot).
location	Either an IP address or a name that resolves to an IP address via the PIX Firewall naming resolution mechanism.
password	The password for logging into the HTTP server.

pdm	Download the selected PDM image files to Flash memory. These files are available to the PIX Firewall immediately, without a reboot.
port	Specifies the port to contact on the HTTP server. It defaults to 80 for **http** and 443 for **https**.
tftp_pathname	PIX Firewall must know how to reach this location via its routing table information. This information is determined by the **ip address** command, the **route** command, or also RIP, depending upon your configuration. The pathname can include any directory names in addition to the actual last component of the path to the file on the server.
user	The username for logging into the HTTP server.

Command Modes

Configuration mode.

Usage Guidelines

copy capture

The **copy capture:** *capture_name* **tftp://***location/path* [**pcap**] command uses the capture name on the PIX Firewall (*capture_name*) as its source and the TFTP address (**tftp://***location/path)* as the copy destination. (These parameters are similar to the **copy tftp** command options.) The addition of the **pcap** option at the end of a **copy capture** command transfers the file in libpcap format.

copy http[s]

The **copy http**[**s**]**://**[*user:password@*] *location* [*:port*] **/** *http_pathname* **flash** [**:** [**image** | **pdm**]] command enables you to download a software image into the Flash memory of the firewall from an HTTP server. SSL is used when the **copy https** command is specified. The *user* and *password* options are used for authentication when logging into the HTTP server. The *location* option is the IP address (or a name that resolves to an IP address) of the HTTP server. The **:***port* option specifies the port on which to contact the server. The value for **:***port* defaults to port 80 for HTTP and port 443 for HTTP through SSL. The *pathname* option is the name of the resource that contains the image or PDM file to copy.

copy tftp

The **copy tftp flash** command enables you to download a software image into the Flash memory of the firewall via TFTP. You can use the **copy tftp flash** command with any PIX Firewall model running Version 5.1 or higher.

The image you download is made available to the PIX Firewall on the next reload (reboot).

The command syntax is as follows:

copy tftp[**:**[[**//***location*][**/***pathname*]]] **flash** [**:**[**image**][**pdm**]]

If the command is used without the *location* or *pathname* optional parameters, then the location and filename are obtained from the user interactively via a series of questions similar to those presented by Cisco IOS software. If you only enter a colon (:), parameters are taken from the **tftp-server** command settings. If other optional parameters are supplied, then these values would be used in place of the corresponding **tftp-server** command setting. Supplying any of the optional parameters, such as a colon and anything after it, causes the command to run without prompting for user input.

The *location* is either an IP address or a name that resolves to an IP address via the PIX Firewall naming resolution mechanism (currently static mappings via the **name** and **names** commands). PIX Firewall must know how to reach this location via its routing table information. This information is determined by the **ip address** command, the **route** command, or also RIP, depending upon your configuration.

The *pathname* can include any directory names besides the actual last component of the path to the file on the server. The pathname cannot contain spaces. If a directory name has spaces, set the directory in the TFTP server instead of in the **copy tftp flash** command.

If your TFTP server has been configured to point to a directory on the system from which you are downloading the image, you need only use the IP address of the system and the image filename.

The TFTP server receives the command and determines the actual file location from its root directory information. The server then downloads the TFTP image to the PIX Firewall.

You can download a TFTP server from the following website:

http://tftpd32.jounin.net

Note Images prior to Version 5.1 cannot be retrieved using this mechanism.

Examples

copy capture

The following example shows the prompts provided when you enter the **copy capture** command without specifying the full path:

```
copy capture:abc tftp
Address or name of remote host [209.165.200.228]?
Source file name [username/cdisk]?
copying capture to tftp://209.165.200.228/username/cdisk:
[yes|no|again]? y
!!!!!!!!!!!!!
```

Alternately, you can specify the full path as follows:

```
copy capture:abc tftp:209.165.200.228/tftpboot/abc.cap pcap
```

If the TFTP server is already configured, the location or file name can be left unspecified as follows:

```
tftp-server outside 209.165.200.228 tftp/cdisk
copy capture:abc tftp:/tftp/abc.cap
```

The following example shows how to use the defaults of the preconfigured TFTP server in the **copy capture** command:

```
copy capture:abc tftp:pcap
```

copy http[s]

The following example shows how to copy the PIX Firewall software image from a public HTTP server into the Flash memory of your PIX Firewall:

```
copy http://209.165.200.228/auto/cdisk flash:image
```

The following example show how to copy the PDM software image through HTTPS (HTTP over SSL), where the SSL authentication is provided by the username `robin` and the password `xyz`:

```
copy https://robin:xyz@209.165.200.228/auto/pdm.bin flash:pdm
```

The following example show how to copy the PIX Firewall software image from an HTTPS server running on a non-standard port, where the file is copied into the software image space in Flash memory by default:

```
copy https://robin:zyx@209.165.200.228:8080/auto/cdisk flash
```

The following examples copy files from 192.133.219.25, which is the IP address for www.cisco.com, to the Flash memory of your PIX Firewall. To use these examples, replace the username and password "cco-username:cco-password" with your CCO username and password. Also note that the URL contains a '?'. To enter this while using the PIX Firewall CLI, it must be preceded by typing Ctrl-v.

To copy PIX Firewall software Version 6.2.2 into the Flash memory of your PIX Firewall from Cisco.com, enter the following command:

```
copy http://cco-username:cco-password@192.133.219.25/cgi-bin/Software/Tablebuild/
download.cgi/pix622.bin?&filename=cisco/ciscosecure/pix/pix622.bin flash:image
```

To copy PDM Version 2.0.2 into the Flash memory of your PIX Firewall from Cisco.com, enter the following command:

```
copy http://cco-username:cco-password@192.133.219.25/cgi-bin/Software/Tablebuild/
download.cgi/pdm-202.bin?&filename=cisco/ciscosecure/pix/pdm-202.bin flash:pdm
```

copy tftp

The following example causes the PIX Firewall to prompt you for the filename and location before you start the TFTP download:

```
copy tftp flash
Address or name of remote host [127.0.0.1]? 10.1.1.5
Source file name [cdisk]? pix512.bin
copying tftp://10.1.1.5/pix512.bin to flash
[yes|no|again]? yes
!!!!!!!!!!!!!!!!!!!!!!!!…
Received 1695744 bytes.
Erasing current image.
Writing 1597496 bytes of image.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!…
Image installed.
```

The next example takes the information from the **tftp-server** command. In this case, the TFTP server is in an intranet and resides on the outside interface. The example sets the filename and location from the **tftp-server** command, saves memory, and then downloads the image to Flash memory.

```
pixfirewall(config)# tftp-server outside 10.1.1.5 pix512.bin
Warning: 'outside' interface has a low security level (0).

pixfirewall(config)# write memory
Building configuration...
Cryptochecksum: 017c452b d54be501 8620ba48 490f7e99
[OK]

pixfirewall(config)# copy tftp: flash
copying tftp://10.1.1.5/pix512.bin to flash
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!…
```

The next example overrides the information in the **tftp-server** command to let you specify alternate information about the filename and location. If you have not set the **tftp-server** command, you can also use the **copy tftp flash** command to specify all information as shown in the second example that follows.

```
copy tftp:/pix512.bin flash
copy tftp://10.0.0.1/pix512.bin flash
```

The next example maps an IP address to the TFTP host name with the **name** command and uses the tftp-host name in the **copy** commands:

```
name 10.1.1.6 tftp-host
copy tftp://tftp-host/pix512.bin flash
copy tftp://tftp-host/tftpboot/pix512.bin flash
```

crashinfo

Configure crash information to write to Flash memory, with the option to force a crash of the firewall.

crashinfo test

crashinfo force [**page-fault | watchdog**]

crashinfo save [**enable | disable**]

no crashinfo save disable

show crashinfo [**save**]

clear crashinfo

Syntax Description

page-fault	Forces a crash of the firewall with a page fault.
save disable	Disables crash information from writing to Flash memory.
save enable	Configures crash information to write to Flash memory. (This is the default behavior.)
test	Tests the firewall's ability to save crash information to Flash memory. This does not actually crash the firewall.
watchdog	Forces a crash of the firewall as a result of watchdogging.

Defaults

By default, the firewall saves the crash information file to Flash memory. In other words, by default the **crashinfo save** command is in your configuration.

Command Modes

The **crashinfo save** commands are available in configuration mode.

The **show crashinfo** commands are available in privileged mode.

Usage Guidelines

The **crashinfo save enable** command does not need to be entered to save crash information to the Flash memory of your firewall; this is the default behavior of the firewall. However, if the firewall unit crashes during start up, the crash information file is not saved, whether or not the **crashinfo save enable** command is in your configuration.The firewall must be fully initialized and running first, and then it can save crash information as it crashes.

The **crashinfo save disable** command turns off saving crash information to the Flash memory of the firewall. After a **crashinfo save disable** command is written to your configuration, crash information is dumped to your console screen only. Use the **crashinfo save enable** or **no crashinfo save disable** command to re-enable saving the crash information to Flash memory.

The **crashinfo test** command provides a simulated crash information file, which it saves to Flash memory. It does not crash the firewall. Use the **crashinfo test** command to test your crash information file configuration without actually having to crash your firewall. However, if a previous crash information file was in Flash memory, the test crash information file overwrites it automatically.

crashinfo force [**page-fault | watchdog**]

Caution Do not use the **crashinfo force** command in a production environment. The **crashinfo force** command truly crashes the firewall and forces it to reload.

The **crashinfo force page-fault** command crashes the firewall as a result of a page fault, and the **crashinfo force watchdog** command crashes the firewall as a result of watchdogging. In the crash output, there is nothing that differentiates a real crash from a crash resulting from the **crashinfo force page-fault** or **crashinfo force watchdog** command (because these are real crashes). The firewall reloads after the crash dump is complete. This command is available only in configuration mode.

If save to crash (**crashinfo save enable**) is enabled then the crash is first dumped to Flash memory and then to the console. Otherwise, it is only dumped to console.

When the **crashinfo force page-fault** command is issued, a warning prompt similar to the following is displayed:

```
pixfirewall(config)# crashinfo force page-fault
WARNING: This command will force the PIX to crash and reboot.
    Do you wish to proceed? [confirm]:
```

If you enter a carriage return (by pressing the return or enter key on your keyboard), "**Y**", or "**y**" the firewall crashes and reloads; all three of these are interpreted as confirmation. Any other character is interpreted as a **no**, and the firewall returns to the command-line configuration mode prompt.

show crashinfo

The **show crashinfo save** command displays whether or not the firewall is currently configured to save crash information to Flash memory.

The **show crashinfo** command displays the crash information file that is stored in the Flash memory of the firewall. If the crash information file is from a test crash (from the **crashinfo test** command), the first string of the crash information file is "**: Saved_Test_Crash**" and the last one is "**: End_Test_Crash**". If the crash information file is from a real crash, the first string of the crash information file is "**: Saved_Crash**" and the last one is "**: End_Crash**" (this includes crashes from use of the **crashinfo force page-fault** or **crashinfo force watchdog** commands).

The **clear crashinfo** command deletes the crash information file from the Flash memory of the firewall.

Examples

The following example shows how to display the current crash information configuration:

```
pixfirewall(config)# show crashinfo save
crashinfo save enable
```

The following example shows the output for a crash information file test. (However, this test does not actually crash the firewall. It provides a simulated example file.)

```
pixfirewall(config)# crashinfo test
pixfirewall(config)# exit
pixfirewall# show crashinfo
: Saved_Test_Crash

Thread Name: ci/console (Old pc 0x001a6ff5 ebp 0x00e88920)

Traceback:
0: 00323143
1: 0032321b
2: 0010885c
3: 0010763c
```

```
4: 001078db
5: 00103585
6: 00000000
    vector 0x000000ff (user defined)
       edi 0x004f20c4
       esi 0x00000000
       ebp 0x00e88c20
       esp 0x00e88bd8
       ebx 0x00000001
       edx 0x00000074
       ecx 0x00322f8b
       eax 0x00322f8b
error code n/a
       eip 0x0010318c
        cs 0x00000008
    eflags 0x00000000
       CR2 0x00000000
Stack dump: base:0x00e8511c size:16384, active:1476
0x00e89118: 0x004f1bb4
0x00e89114: 0x001078b4
0x00e89110-0x00e8910c: 0x00000000
0x00e89108-0x00e890ec: 0x12345678
0x00e890e8: 0x004f1bb4
0x00e890e4: 0x00103585
0x00e890e0: 0x00e8910c
0x00e890dc-0x00e890cc: 0x12345678
0x00e890c8: 0x00000000
0x00e890c4-0x00e890bc: 0x12345678
0x00e890b8: 0x004f1bb4
0x00e890b4: 0x001078db
0x00e890b0: 0x00e890e0
0x00e890ac-0x00e890a8: 0x12345678
0x00e890a4: 0x001179b3
0x00e890a0: 0x00e890b0
0x00e8909c-0x00e89064: 0x12345678
0x00e89060: 0x12345600
0x00e8905c: 0x20232970
0x00e89058: 0x616d2d65
0x00e89054: 0x74002023
0x00e89050: 0x29676966
0x00e8904c: 0x6e6f6328
0x00e89048: 0x31636573
0x00e89044: 0x7069636f
0x00e89040: 0x64786970
0x00e8903c-0x00e88e50: 0x00000000
0x00e88e4c: 0x000a7473
0x00e88e48: 0x6574206f
0x00e88e44: 0x666e6968
0x00e88e40: 0x73617263
0x00e88e3c-0x00e88e38: 0x00000000
0x00e88e34: 0x12345600
0x00e88e30-0x00e88dfc: 0x00000000
0x00e88df8: 0x00316761
0x00e88df4: 0x74706100
0x00e88df0: 0x12345600
0x00e88dec-0x00e88ddc: 0x00000000
0x00e88dd8: 0x00000070
0x00e88dd4: 0x616d2d65
0x00e88dd0: 0x74756f00
0x00e88dcc: 0x00000000
0x00e88dc8: 0x00e88e40
0x00e88dc4: 0x004f20c4
0x00e88dc0: 0x12345600
0x00e88dbc: 0x00000000
```

```
0x00e88db8: 0x00000035
0x00e88db4: 0x315f656c
0x00e88db0: 0x62616e65
0x00e88dac: 0x0030fcf0
0x00e88da8: 0x3011111f
0x00e88da4: 0x004df43c
0x00e88da0: 0x0053fef0
0x00e88d9c: 0x004f1bb4
0x00e88d98: 0x12345600
0x00e88d94: 0x00000000
0x00e88d90: 0x00000035
0x00e88d8c: 0x315f656c
0x00e88d88: 0x62616e65
0x00e88d84: 0x00000000
0x00e88d80: 0x004f20c4
0x00e88d7c: 0x00000001
0x00e88d78: 0x01345678
0x00e88d74: 0x00f53854
0x00e88d70: 0x00f7f754
0x00e88d6c: 0x00e88db0
0x00e88d68: 0x00e88d7b
0x00e88d64: 0x00f53874
0x00e88d60: 0x00e89040
0x00e88d5c-0x00e88d54: 0x12345678
0x00e88d50-0x00e88d4c: 0x00000000
0x00e88d48: 0x004f1bb4
0x00e88d44: 0x00e88d7c
0x00e88d40: 0x00e88e40
0x00e88d3c: 0x00f53874
0x00e88d38: 0x004f1bb4
0x00e88d34: 0x0010763c
0x00e88d30: 0x00e890b0
0x00e88d2c: 0x00e88db0
0x00e88d28: 0x00e88d88
0x00e88d24: 0x0010761a
0x00e88d20: 0x00e890b0
0x00e88d1c: 0x00e88e40
0x00e88d18: 0x00f53874
0x00e88d14: 0x0010166d
0x00e88d10: 0x0000000e
0x00e88d0c: 0x00f53874
0x00e88d08: 0x00f53854
0x00e88d04: 0x0048b301
0x00e88d00: 0x00e88d30
0x00e88cfc: 0x0000000e
0x00e88cf8: 0x00f53854
0x00e88cf4: 0x0048a401
0x00e88cf0: 0x00f53854
0x00e88cec: 0x00f53874
0x00e88ce8: 0x0000000e
0x00e88ce4: 0x0048a64b
0x00e88ce0: 0x0000000e
0x00e88cdc: 0x00f53874
0x00e88cd8: 0x00f7f96c
0x00e88cd4: 0x0048b4f8
0x00e88cd0: 0x00e88d00
0x00e88ccc: 0x0000000f
0x00e88cc8: 0x00f7f96c
0x00e88cc4-0x00e88cc0: 0x0000000e
0x00e88cbc: 0x00e89040
0x00e88cb8: 0x00000000
0x00e88cb4: 0x00f5387e
0x00e88cb0: 0x00f53874
0x00e88cac: 0x00000002
```

```
0x00e88ca8: 0x00000001
0x00e88ca4: 0x00000009
0x00e88ca0-0x00e88c9c: 0x00000001
0x00e88c98: 0x00e88cb0
0x00e88c94: 0x004f20c4
0x00e88c90: 0x0000003a
0x00e88c8c: 0x00000000
0x00e88c88: 0x0000000a
0x00e88c84: 0x00489f3a
0x00e88c80: 0x00e88d88
0x00e88c7c: 0x00e88e40
0x00e88c78: 0x00e88d7c
0x00e88c74: 0x001087ed
0x00e88c70: 0x00000001
0x00e88c6c: 0x00e88cb0
0x00e88c68: 0x00000002
0x00e88c64: 0x0010885c
0x00e88c60: 0x00e88d30
0x00e88c5c: 0x00727334
0x00e88c58: 0xa0ffffff
0x00e88c54: 0x00e88cb0
0x00e88c50: 0x00000001
0x00e88c4c: 0x00e88cb0
0x00e88c48: 0x00000002
0x00e88c44: 0x0032321b
0x00e88c40: 0x00e88c60
0x00e88c3c: 0x00e88c7f
0x00e88c38: 0x00e88c5c
0x00e88c34: 0x004b1ad5
0x00e88c30: 0x00e88c60
0x00e88c2c: 0x00e88e40
0x00e88c28: 0xa0ffffff
0x00e88c24: 0x00323143
0x00e88c20: 0x00e88c40
0x00e88c1c: 0x00000000
0x00e88c18: 0x00000008
0x00e88c14: 0x0010318c
0x00e88c10-0x00e88c0c: 0x00322f8b
0x00e88c08: 0x00000074
0x00e88c04: 0x00000001
0x00e88c00: 0x00e88bd8
0x00e88bfc: 0x00e88c20
0x00e88bf8: 0x00000000
0x00e88bf4: 0x004f20c4
0x00e88bf0: 0x000000ff
0x00e88bec: 0x00322f87
0x00e88be8: 0x00f5387e
0x00e88be4: 0x00323021
0x00e88be0: 0x00e88c10
0x00e88bdc: 0x004f20c4
0x00e88bd8: 0x00000000 *
0x00e88bd4: 0x004eabb0
0x00e88bd0: 0x00000001
0x00e88bcc: 0x00f5387e
0x00e88bc8-0x00e88bc4: 0x00000000
0x00e88bc0: 0x00000008
0x00e88bbc: 0x0010318c
0x00e88bb8-0x00e88bb4: 0x00322f8b
0x00e88bb0: 0x00000074
0x00e88bac: 0x00000001
0x00e88ba8: 0x00e88bd8
0x00e88ba4: 0x00e88c20
0x00e88ba0: 0x00000000
0x00e88b9c: 0x004f20c4
```

```
0x00e88b98: 0x000000ff
0x00e88b94: 0x001031f2
0x00e88b90: 0x00e88c20
0x00e88b8c: 0xffffffff
0x00e88b88: 0x00e88cb0
0x00e88b84: 0x00320032
0x00e88b80: 0x37303133
0x00e88b7c: 0x312f6574
0x00e88b78: 0x6972772f
0x00e88b74: 0x342f7665
0x00e88b70: 0x64736666
0x00e88b6c: 0x00020000
0x00e88b68: 0x00000010
0x00e88b64: 0x00000001
0x00e88b60: 0x123456cd
0x00e88b5c: 0x00000000
0x00e88b58: 0x00000008

Cisco PIX Firewall Version 6.3
Cisco PIX Device Manager Version 2.1

Compiled on Fri 15-Nov-02 14:35 by root

pixfirewall up 10 days 0 hours

Hardware:   PIX-515, 64 MB RAM, CPU Pentium 200 MHz
Flash i28F640J5 @ 0x300, 16MB
BIOS Flash AT29C257 @ 0xfffd8000, 32KB

0: ethernet0: address is 0003.e300.73fd, irq 10
1: ethernet1: address is 0003.e300.73fe, irq 7
2: ethernet2: address is 00d0.b7c8.139e, irq 9
Licensed Features:
Failover:           Disabled
VPN-DES:            Enabled
VPN-3DES-AES:       Disabled
Maximum Interfaces: 3
Cut-through Proxy:  Enabled
Guards:             Enabled
URL-filtering:      Enabled
Inside Hosts:       Unlimited
Throughput:         Unlimited
IKE peers:          Unlimited

This PIX has a Restricted (R) license.

Serial Number: 480430455 (0x1ca2c977)
Running Activation Key: 0xc2e94182 0xc21d8206 0x15353200 0x633f6734
Configuration last modified by enable_15 at 13:49:42.148 UTC Wed Nov 20 2002

------------------ show clock ------------------

15:34:28.129 UTC Sun Nov 24 2002

------------------ show memory ------------------

Free memory:        50444824 bytes
Used memory:        16664040 bytes
-------------     ----------------
Total memory:       67108864 bytes

------------------ show conn count ------------------

0 in use, 0 most used
```

```
------------------ show xlate count ------------------

0 in use, 0 most used

------------------ show blocks ------------------

  SIZE    MAX    LOW    CNT
     4   1600   1600   1600
    80    400    400    400
   256    500    499    500
  1550   1188    795    927

------------------ show interface ------------------

interface ethernet0 "outside" is up, line protocol is up
  Hardware is i82559 ethernet, address is 0003.e300.73fd
  IP address 172.23.59.232, subnet mask 255.255.0.0
  MTU 1500 bytes, BW 10000 Kbit half duplex
        6139 packets input, 830375 bytes, 0 no buffer
        Received 5990 broadcasts, 0 runts, 0 giants
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
        90 packets output, 6160 bytes, 0 underruns
        0 output errors, 13 collisions, 0 interface resets
        0 babbles, 0 late collisions, 47 deferred
        0 lost carrier, 0 no carrier
        input queue (curr/max blocks): hardware (5/128) software (0/2)
        output queue (curr/max blocks): hardware (0/1) software (0/1)
interface ethernet1 "inside" is up, line protocol is down
  Hardware is i82559 ethernet, address is 0003.e300.73fe
  IP address 10.1.1.1, subnet mask 255.255.255.0
  MTU 1500 bytes, BW 10000 Kbit half duplex
        0 packets input, 0 bytes, 0 no buffer
        Received 0 broadcasts, 0 runts, 0 giants
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
        1 packets output, 60 bytes, 0 underruns
        0 output errors, 0 collisions, 0 interface resets
        0 babbles, 0 late collisions, 0 deferred
        1 lost carrier, 0 no carrier
        input queue (curr/max blocks): hardware (128/128) software (0/0)
        output queue (curr/max blocks): hardware (0/1) software (0/1)
interface ethernet2 "intf2" is administratively down, line protocol is down
  Hardware is i82559 ethernet, address is 00d0.b7c8.139e
  IP address 127.0.0.1, subnet mask 255.255.255.255
  MTU 1500 bytes, BW 10000 Kbit half duplex
        0 packets input, 0 bytes, 0 no buffer
        Received 0 broadcasts, 0 runts, 0 giants
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
        0 packets output, 0 bytes, 0 underruns
        0 output errors, 0 collisions, 0 interface resets
        0 babbles, 0 late collisions, 0 deferred
        0 lost carrier, 0 no carrier
        input queue (curr/max blocks): hardware (128/128) software (0/0)
        output queue (curr/max blocks): hardware (0/0) software (0/0)

------------------ show cpu usage ------------------

CPU utilization for 5 seconds = 0%; 1 minute: 0%; 5 minutes: 0%

------------------ show process ------------------


    PC       SP       STATE       Runtime    SBASE     Stack Process
Hsi 001e3329 00763e7c 0053e5c8          0 00762ef4 3784/4096 arp_timer
```

```
Lsi 001e80e9 00807074 0053e5c8          0 008060fc 3792/4096 FragDBGC
Lwe 00117e3a 009dc2e4 00541d18          0 009db46c 3704/4096 dbgtrace
Lwe 003cee95 009de464 00537718          0 009dc51c 8008/8192 Logger
Hwe 003d2d18 009e155c 005379c8          0 009df5e4 8008/8192 tcp_fast
Hwe 003d2c91 009e360c 005379c8          0 009e1694 8008/8192 tcp_slow
Lsi 002ec97d 00b1a464 0053e5c8          0 00b194dc 3928/4096 xlate clean
Lsi 002ec88b 00b1b504 0053e5c8          0 00b1a58c 3888/4096 uxlate clean
Mrd 002e3a17 00c8f8d4 0053e600          0 00c8d93c 7908/8192 tcp_intercept_times
Lsi 00423dd5 00d3a22c 0053e5c8          0 00d392a4 3900/4096 route_process
Hsi 002d59fc 00d3b2bc 0053e5c8          0 00d3a354 3780/4096 PIX Garbage Collecr
Hwe 0020e301 00d5957c 0053e5c8          0 00d55614 16048/16384 isakmp_time_keepr
Lsi 002d377c 00d7292c 0053e5c8          0 00d719a4 3928/4096 perfmon
Hwe 0020bd07 00d9c12c 0050bb90          0 00d9b1c4 3944/4096 IPSec
Mwe 00205e25 00d9e1ec 0053e5c8          0 00d9c274 7860/8192 IPsec timer handler
Hwe 003864e3 00db26bc 00557920          0 00db0764 6904/8192 qos_metric_daemon
Mwe 00255a65 00dc9244 0053e5c8          0 00dc8adc 1436/2048 IP Background
Lwe 002e450e 00e7bb94 00552c30          0 00e7ad1c 3704/4096 pix/trace
Lwe 002e471e 00e7cc44 00553368          0 00e7bdcc 3704/4096 pix/tconsole
Hwe 001e5368 00e7ed44 00730674          0 00e7ce9c 7228/8192 pix/intf0
Hwe 001e5368 00e80e14 007305d4          0 00e7ef6c 7228/8192 pix/intf1
Hwe 001e5368 00e82ee4 00730534       2470 00e8103c 4892/8192 pix/intf2
H*  001a6ff5 0009ff2c 0053e5b0       4820 00e8511c 12860/16384 ci/console
Csi 002dd8ab 00e8a124 0053e5c8          0 00e891cc 3396/4096 update_cpu_usage
Hwe 002cb4d1 00f2bfbc 0051e360          0 00f2a134 7692/8192 uauth_in
Hwe 003d17d1 00f2e0bc 00828cf0          0 00f2c1e4 7896/8192 uauth_thread
Hwe 003e71d4 00f2f20c 00537d20          0 00f2e294 3960/4096 udp_timer
Hsi 001db3ca 00f30fc4 0053e5c8          0 00f3004c 3784/4096 557mcfix
Crd 001db37f 00f32084 0053ea40  508286220 00f310fc 3688/4096 557poll
Lsi 001db435 00f33124 0053e5c8          0 00f321ac 3700/4096 557timer
Hwe 001e5398 00f441dc 008121e0          0 00f43294 3912/4096 fover_ip0
Cwe 001dcdad 00f4523c 00872b48        120 00f44344 3528/4096 ip/0:0
Hwe 001e5398 00f4633c 008121bc         10 00f453f4 3532/4096 icmp0
Hwe 001e5398 00f47404 00812198          0 00f464cc 3896/4096 udp_thread/0
Hwe 001e5398 00f4849c 00812174          0 00f475a4 3456/4096 tcp_thread/0
Hwe 001e5398 00f495bc 00812150          0 00f48674 3912/4096 fover_ip1
Cwe 001dcdad 00f4a61c 008ea850          0 00f49724 3832/4096 ip/1:1
Hwe 001e5398 00f4b71c 0081212c          0 00f4a7d4 3912/4096 icmp1
Hwe 001e5398 00f4c7e4 00812108          0 00f4b8ac 3896/4096 udp_thread/1
Hwe 001e5398 00f4d87c 008120e4          0 00f4c984 3832/4096 tcp_thread/1
Hwe 001e5398 00f4e99c 008120c0          0 00f4da54 3912/4096 fover_ip2
Cwe 001e542d 00f4fa6c 00730534          0 00f4eb04 3944/4096 ip/2:2
Hwe 001e5398 00f50afc 0081209c          0 00f4fbb4 3912/4096 icmp2
Hwe 001e5398 00f51bc4 00812078          0 00f50c8c 3896/4096 udp_thread/2
Hwe 001e5398 00f52c5c 00812054          0 00f51d64 3832/4096 tcp_thread/2
Hwe 003d1a65 00f78284 008140f8          0 00f77fdc  300/1024 listen/http1
Mwe 0035cafa 00f7a63c 0053e5c8          0 00f786c4 7640/8192 Crypto CA

------------------ show failover ------------------

No license for Failover

------------------ show traffic ------------------

outside:
        received (in 865565.090 secs):
                6139 packets    830375 bytes
                0 pkts/sec      0 bytes/sec
        transmitted (in 865565.090 secs):
                90 packets      6160 bytes
                0 pkts/sec      0 bytes/sec
inside:
        received (in 865565.090 secs):
                0 packets       0 bytes
                0 pkts/sec      0 bytes/sec
```

```
        transmitted (in 865565.090 secs):
                1 packets       60 bytes
                0 pkts/sec      0 bytes/sec
intf2:
        received (in 865565.090 secs):
                0 packets       0 bytes
                0 pkts/sec      0 bytes/sec
        transmitted (in 865565.090 secs):
                0 packets       0 bytes
                0 pkts/sec      0 bytes/sec

------------------ show perfmon ------------------


PERFMON STATS:    Current      Average
Xlates               0/s          0/s
Connections          0/s          0/s
TCP Conns            0/s          0/s
UDP Conns            0/s          0/s
URL Access           0/s          0/s
URL Server Req       0/s          0/s
TCP Fixup            0/s          0/s
TCPIntercept         0/s          0/s
HTTP Fixup           0/s          0/s
FTP Fixup            0/s          0/s
AAA Authen           0/s          0/s
AAA Author           0/s          0/s
AAA Account          0/s          0/s
: End_Test_Crash
```

Related Commands		
	failover	Enable or disable the PIX Firewall failover feature on a standby PIX Firewall.

crypto dynamic-map

Create, view, or delete a dynamic crypto map entry.

[**no**] **crypto dynamic-map** *dynamic-map-name dynamic-seq-num* **match address** *acl_name*

[**no**] **crypto dynamic-map** *dynamic-map-name dynamic-seq-num* **set peer** *hostname* | *ip_address*

[**no**] **crypto dynamic-map** *dynamic-map-name dynamic-seq-num* **set pfs** [**group1** | **group2**]

[**no**] **crypto dynamic-map** *dynamic-map-name dynamic-seq-num* **set security-association lifetime seconds** *seconds* | **kilobytes** *kilobytes*

[**no**] **crypto dynamic-map** *dynamic-map-name dynamic-seq-num* **set transform-set** *transform-set-name1* [*... transform-set-name9*]

clear [**crypto**] **dynamic-map** [*dynamic-map-name*] [*dynamic-seq-num*]

show crypto dynamic-map [**tag** *dynamic-map-name*]

Syntax Description

dynamic-map-name	Specify the name of the dynamic crypto map set.
dynamic-seq-num	Specify the sequence number that corresponds to the dynamic crypto map entry.
subcommand	Various subcommands (**match address**, **set transform-set**, and so on).
tag *map-name*	(Optional) Show the crypto dynamic map set with the specified *map-name*.

Note The **crypto dynamic-map** subcommands, such as **match address**, **set peer**, and **set pfs** are described with the **crypto map** command. If the peer initiates the negotiation and the local configuration specifies perfect forward secrecy (PFS), the peer must perform a PFS exchange or the negotiation will fail. If the local configuration does not specify a group, a default of group1 will be assumed, and an offer of either group1 or group2 will be accepted. If the local configuration specifies group2, that group must be part of the peer's offer or the negotiation will fail. If the local configuration does not specify PFS, it will accept any offer of PFS from the peer.

Command Modes

Configuration mode.

Usage Guidelines

The sections that follow describe each **crypto dynamic-map** command.

crypto dynamic-map

The **crypto dynamic-map** command lets you create a dynamic crypto map entry. The **no crypto dynamic-map** command deletes a dynamic crypto map set or entry. The **clear** [**crypto**] **dynamic-map r**emoves all of the dynamic crypto map command statements. Specifying the name of a given crypto dynamic map removes the associated crypto dynamic map command statement(s). You can also specify the dynamic crypto map's sequence number to remove all of the associated dynamic crypto map command statements. The **show crypto dynamic-map** command lets you view a dynamic crypto map set.

Dynamic crypto maps are policy templates used when processing negotiation requests for new security associations from a remote IPSec peer, even if you do not know all of the crypto map parameters required to communicate with the peer (such as the peer's IP address). For example, if you do not know about all the remote IPSec peers in your network, a dynamic crypto map lets you accept requests for new security associations from previously unknown peers. (However, these requests are not processed until the IKE authentication has completed successfully.)

When a PIX Firewall receives a negotiation request via IKE from another peer, the request is examined to see if it matches a crypto map entry. If the negotiation does not match any explicit crypto map entry, it will be rejected unless the crypto map set includes a reference to a dynamic crypto map.

The dynamic crypto map accepts "wildcard" parameters for any parameters not explicitly stated in the dynamic crypto map entry. This lets you set up IPSec security associations with a previously unknown peer. (The peer still must specify matching values for the "wildcard" IPSec security association negotiation parameters.)

If the PIX Firewall 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 PIX Firewall 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 removed.

The **crypto dynamic-map** command statements are used for determining whether or not traffic should be protected. The only parameter required in a **crypto dynamic-map** command statement is the **set transform-set**. All other parameters are optional.

Examples

The following example configures an IPSec crypto map set:

Crypto map entry **mymap 30** references the dynamic crypto map set **mydynamicmap**, which can be used to process inbound security association negotiation requests that do not match **mymap** entries 10 or 20. In this case, if the peer specifies a transform set that matches one of the transform sets specified in **mydynamicmap**, for a flow "permitted" by the access list 103, IPSec will accept the request and set up security associations with the remote peer without previously knowing about the peer. If accepted, the resulting security associations (and temporary crypto map entry) are established according to the settings specified by the remote peer.

The access list associated with **mydynamicmap 10** is also used as a filter. Inbound packets that match a permit statement in this list are dropped for not being IPSec protected. (The same is true for access lists associated with static crypto maps entries.) Outbound packets that match a permit statement without an existing corresponding IPSec security association are also dropped in the following example.

```
crypto map mymap 10 ipsec-isakmp
crypto map mymap 10 match address 101
crypto map mymap 10 set transform-set my_t_set1
crypto map mymap 10 set peer 10.0.0.1 10.0.0.2
crypto map mymap 20 ipsec-isakmp
crypto map mymap 20 match address 102
crypto map mymap 20 set transform-set my_t_set1 my_t_set2
crypto map mymap 20 set peer 10.0.0.3
crypto dynamic-map mydynamicmap 10 match address 103

crypto dynamic-map mydynamicmap 10 set transform-set my_t_set1 my_t_set2 my_t_set3

crypto map mymap 30 ipsec-isakmp dynamic mydynamicmap
```

The following is sample output from the **how crypto dynamic-map** command:

```
show crypto dynamic-map

Crypto Map Template "dyn1" 10

        access-list 152 permit ip host 172.21.114.67 any
        Current peer: 0.0.0.0
        Security association lifetime: 4608000 kilobytes/120 seconds
        PFS (Y/N): N
        Transform sets={ tauth, t1, }
```

The following partial configuration was in effect when the preceding **show crypto dynamic-map** command was issued:

```
crypto ipsec security-association lifetime seconds 120
crypto ipsec transform-set t1 esp-des esp-md5-hmac
crypto ipsec transform-set tauth ah-sha-hmac
crypto dynamic-map dyn1 10 set transform-set tauth t1
crypto dynamic-map dyn1 10 match address 152
crypto map to-firewall local-address Ethernet0
crypto map to-firewall 10 ipsec-isakmp
crypto map to-firewall 10 set peer 172.21.114.123
```

```
crypto map to-firewall 10 set transform-set tauth t1
crypto map to-firewall 10 match address 150
crypto map to-firewall 20 ipsec-isakmp dynamic dyn1
access-list 150 permit ip host 172.21.114.67 host 172.21.114.123
access-list 150 permit ip host 15.15.15.1 host 172.21.114.123
access-list 150 permit ip host 15.15.15.1 host 8.8.8.1
access-list 152 permit ip host 172.21.114.67 any
```

The following example shows output from the **show crypto map** command for a crypto map named "mymap":

```
pixfirewall(config)# show crypto map

Crypto Map: "mymap" interfaces: { outside }

Crypto Map "mymap" 1 ipsec-isakmp
        Peer = 209.165.200.241
        access-list no-nat; 1 elements
        access-list no-nat permit ip 209.165.201.16 255.255.255.0 1.1.1.0 255.255.255.0
(hitcnt=0)
        Current peer: 209.165.200.241
        Security association lifetime: 4608000 kilobytes/28800 seconds
        PFS (Y/N): Y
        DH group:  group5
        Transform sets={ mycrypt, }
```

crypto dynamic-map match address

See the **crypto map match address** command within the **crypto map** command for information about this command.

crypto dynamic-map set peer

See the **crypto map set peer** command within the **crypto map** command for information about this command.

crypto dynamic-map set pfs

See the **crypto map set pfs** command within the **crypto map** command for information about this command.

crypto dynamic-map set security-association lifetime

See the **crypto map set security-association lifetime** command within the **crypto map** command for information about this command.

crypto dynamic-map set transform-set

See the **crypto map set transform-set** command within the **crypto map** command for information about this command.

Note The **crypto map set transform-set** command is required for dynamic crypto map entries.

crypto ipsec

Create, view, or delete IPSec security associations, security association global lifetime values, and global transform sets.

[**no**] **crypto ipsec security-association lifetime seconds** *seconds* | **kilobytes** *kilobytes*

crypto ipsec transform-set *transform-set-name transform1* [*transform2* [*transform3*]]

crypto ipsec transform-set *transform-set-name* **mode transport**

[**no**] **crypto ipsec transform-set trans-name** [**ah-md5-hmac | ah-sha-hmac**] [**esp-aes |esp-aes-192 | esp-aes-256| esp-des | esp-3des| esp-null**] [**esp-md5-hmac | esp-sha-hmac**]

clear [**crypto**] **ipsec sa**

clear [**crypto**] **ipsec sa counters**

clear [**crypto**] **ipsec sa entry** *destination-address protocol spi*

clear [**crypto**] **ipsec sa map** *map-name*

clear [**crypto**] **ipsec sa peer**

show crypto ipsec security-association lifetime

show crypto ipsec transform-set [**tag** *transform-set-name*]

show crypto ipsec sa [**map** *map-name* | **address** | **identity**] [**detail**]

Syntax Description

address	(Optional) Show all of the existing security associations, sorted by the destination address (either the local address or the address of the remote IPSec peer) and then by protocol (AH or ESP).
esp-aes	Selecting this option means that IPSec messages protected by this transform are encrypted using AES with a 128-bit key.
esp-aes-192	Selecting this option means that IPSec messages protected by this transform are encrypted using AES with a 192-bit key.
esp-aes-256	Selecting this option means that IPSec messages protected by this transform are encrypted using AES with a 256-bit key.
destination-address	Specify the IP address of your peer or the remote peer.
detail	(Optional) Show detailed error counters.
identity	(Optional) Show only the flow information. It does not show the security association information.
kilobytes *kilobytes*	Specify the volume of traffic (in kilobytes) that can pass between IPSec peers using a given security association before that security association expires. The default is 4,608,000 kilobytes (10 megabytes per second for one hour).
map *map-name*	The name of the crypto map set.
mode *transport*	Specifies the transform set to accept transport mode requests in addition to the tunnel mode request.
protocol	Specify either the AH or ESP protocol.

seconds *seconds*	Specify the number of seconds a security association will live before it expires. The default is 28,800 seconds (eight hours).
seq-num	The number you assign to the crypto map entry.
spi	Specify the Security Parameter Index (SPI), a number that is used to uniquely identify a security association. The SPI is an arbitrary number you assign in the range of 256 to 4,294,967,295 (a hexidecimal value of FFFF FFFF).
tag *transform-set-name*	(Optional) Show only the transform sets with the specified *transform-set-name*.
transform1 *transform2* *transform3*	Specify up to three transforms. Transforms define the IPSec security protocol(s) and algorithm(s). Each transform represents an IPSec security protocol (ESP, AH, or both) plus the algorithm you want to use.
transform-set-name	Specify the name of the transform set to create or modify.

Command Modes

Configuration mode.

Usage Guidelines

The sections that follow describe each **crypto ipsec** command. To run the Known Answer Test (KAT), refer to the **show crypto engine verify** command.

crypto ipsec security-association lifetime

The **crypto ipsec security-association lifetime** command is used to change global lifetime values used when negotiating IPSec security associations. To reset a lifetime to the default value, use the **no crypto ipsec security-association lifetime** command. The **show crypto ipsec security-association lifetime** command lets you view the security-association lifetime value configured for a particular crypto map entry.

IPSec security associations use shared secret keys. These keys and their security associations time out together.

Assuming that the particular crypto map entry does not have lifetime values configured, when the PIX Firewall requests new security associations during security association negotiation, it will specify its global lifetime value in the request to the peer; it will use this value as the lifetime of the new security associations. When the PIX Firewall receives a negotiation request from the peer, it will use the smaller of the lifetime value proposed by the peer or the locally configured lifetime value as the lifetime of the new security associations.

There are two lifetimes: a “timed” lifetime and a “traffic-volume” lifetime. The security association expires after the first of these lifetimes is reached.

If you change a global lifetime, the change is only applied when the crypto map entry does not have a lifetime value specified. 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**] **ipsec sa** command. See the **clear** [**crypto**] **ipsec sa** command for more information.

To change the global timed lifetime, use the **crypto ipsec security-association lifetime seconds** command. The timed lifetime causes the security association to time out after the specified number of seconds have passed.

To change the global traffic-volume lifetime, use the **crypto ipsec security-association lifetime kilobytes** command. The traffic-volume lifetime causes the security association to time out after the specified amount of traffic (in kilobytes) has been protected by the security associations' key.

Shorter lifetimes can make it harder to mount a successful key recovery attack, because the attacker has less data encrypted under the same key to work with. However, shorter lifetimes require more CPU processing time for establishing new security associations. The lifetime values are ignored for manually established security associations (security associations installed using an **ipsec-manual crypto map** command entry).

The security association (and corresponding keys) will expire according to whichever occurs sooner, either after the number of seconds has passed (specified by the **seconds** keyword) or after the amount of traffic in kilobytes has passed (specified by the **kilobytes** keyword).

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 occurs 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.

clear [crypto] ipsec sa

Use the **clear** [**crypto**] **ipsec sa** command to delete IPSec security associations. The keyword **crypto** is optional. If the security associations were established via IKE, they are deleted and future IPSec traffic will require new security associations to be negotiated. When IKE is used, the IPSec security associations are established only when needed.

If the security associations are manually established, the security associations are deleted.

If the **peer**, **map**, **entry**, or **counters** keywords are not used, all IPSec security associations will be deleted. This command clears (deletes) IPSec security associations.

If the security associations were established via IKE, they are deleted and future IPSec traffic will require new security associations to be negotiated. (When IKE is used, the IPSec security associations are established only when needed.)

If the security associations are manually established, the security associations are deleted and reinstalled. (When IKE is not used, the IPSec security associations are created as soon as the configuration is completed.)

If the **peer**, **map**, **entry**, o**r counters** keywords are not used, all IPSec security associations will be deleted.

The **peer** keyword deletes any IPSec security associations for the specified peer.

The **map** keyword deletes any IPSec security associations for the named crypto map set.

The **entry** keyword deletes the IPSec security association with the specified address, protocol, and SPI.

If any of the previous commands cause a particular security association to be deleted, all the “sibling” security associations—that were established during the same IKE negotiation—are deleted as well.

The **counters** keyword simply clears the traffic counters maintained for each security association; it does not clear the security associations themselves.

If you make configuration changes that affect security associations, these changes will not apply to existing security associations but to negotiations for subsequent security associations. You can use the **clear [crypto] ipsec sa** command to restart all security associations so they will use the most current configuration settings. In the case of manually established security associations, if you make changes that affect security associations you must use the **clear [crypto] ipsec sa** command before the changes take effect.

Note If you make significant changes to an IPSec configuration, such as to access lists or peers, the **clear** [**crypto**] **ipsec sa** command does not enable the new configuration. In such a case, rebind the crypto map to the interface with the **crypto map interface** command.

If the PIX Firewall is processing active IPSec traffic, we recommend that you only clear the portion of the security association database that is affected by the changes to avoid causing active IPSec traffic to temporarily fail.

The **clear** [**crypto**] **ipsec sa** command only clears IPSec security associations; to clear IKE security associations, use the **clear** [**crypto**] **isakmp sa** command.

The following example clears (and reinitializes if appropriate) all IPSec security associations at the PIX Firewall:

```
clear crypto ipsec sa
```

The following example clears (and reinitializes if appropriate) the inbound and outbound IPSec security associations established along with the security association established for address 10.0.0.1 using the AH protocol with the SPI of 256:

```
clear crypto ipsec sa entry 10.0.0.1 AH 256
```

show crypto ipsec sa

The **show crypto ipsec sa** command lets you view the settings used by current security associations. If no keyword is used, all security associations are displayed. They are sorted first by interface, and then by traffic flow (for example, source/destination address, mask, protocol, port). Within a flow, the security associations are listed by protocol (ESP/AH) and direction (inbound/outbound).

Note While entering the **show crypto ipsec sa** command, if the screen display is stopped with the More prompt and the security association lifetime expires while the screen display is stopped, then the subsequent display information may refer to a stale security association. Assume that the security association lifetime values that display are invalid.

Output from the **show crypto ipsec sa** command lists the PCP protocol. This is a compression protocol supplied with the Cisco IOS software code on which the PIX Firewall IPSec implementation is based; however, the PIX Firewall does not support the PCP protocol.

crypto ipsec transform-set *transform-set-name* mode transport

This command specifies IPSec **transport** mode for a transform set. The Windows 2000 L2TP/IPSec client uses IPSec transport mode, so **transport** mode must be selected on the transform set. The default is tunnel mode. For PIX Firewall Version 6.0 and higher, L2TP is the only protocol that can use the IPSec transport mode. All other types of packets using IPSec transport mode will be discarded by the PIX Firewall. Use the **no** form of the command to reset the mode to the default value of tunnel mode.

Note A transport mode transform can only be used on a **dynamic** crypto map, and the PIX Firewall CLI will display an error if you attempt to tie a transport-mode transform to a **static** crypto map.

Tunnel mode is automatically enabled for a transform set, so no **mode** needs to be explicitly configured when tunnel mode is desired.

The firewall uses tunnel mode except when it is talking to a Windows 2000 L2TP/IPSec client, with which it uses transport mode. Use the **crypto ipsec transform-set** *trans_name* **mode transport** command to configure the firewall to negotiate with a Windows 2000 L2TP/IPSec client. To reset the **mode** to the default value of tunnel mode, use the **no crypto ipsec transform-set** *trans_name* **mode transport** command.

The **crypto ipsec transform-set** command defines a transform set. To delete a transform set, use the **no crypto ipsec transform-set** command. To view the configured transform sets, use the **show crypto ipsec transform-set command.**

A transform set specifies one or two IPSec security protocols (either ESP or AH or both) and specifies which algorithms to use with the selected security protocol. During the IPSec security association negotiation, the peers agree to use a particular transform set when protecting a particular data flow.

IPSec messages can be protected by a transform set using AES with a 128-bit key, 192-bit key, or 256-bit key.

The following example uses the AES 192-bit key transform:

```
pixfirewall(config)# crypto ipsec transform-set standard esp-aes-192 esp-md5-hmac
```

Note AES support is available on firewalls licensed for VPN-3DES only.

Due to the large key sizes provided by AES, ISAKMP negotiation should use Diffie-Hellman group 5 instead of group 1 or group 2. This is done with the **isakmp policy priority group 5** command.

You can configure 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 is used in the IPSec security association negotiation to protect the data flows specified by that crypto map entry's access list. During the negotiation, 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 is applied to the protected traffic as part of both peer's IPSec security associations.

When security associations are established manually, a single transform set must be used. The transform set is not negotiated.

Before a transform set can be included in a crypto map entry, it must be defined using the **crypto ipsec transform-set** command.

To define a transform set, you specify one to three "transforms"—each transform represents an IPSec security protocol (ESP or AH) plus the algorithm you want to use. When the particular transform set is used during negotiations for IPSec security associations, the entire transform set (the combination of protocols, algorithms, and other settings) must match a transform set at the remote peer.

In a transform set you can specify the AH protocol or the ESP protocol. If you specify an ESP protocol in a transform set, you can specify just an ESP encryption transform or both an ESP encryption transform and an ESP authentication transform.

Examples of acceptable transform combinations are as follows:

- **ah-md5-hmac**
- **esp-des**
- **esp-des** and **esp-md5-hmac**
- **ah-sha-hmac** and **esp-des** and **esp-sha-hmac**

If one or more transforms are specified in the **crypto ipsec transform-set** command for an existing transform set, the specified transforms will replace the existing transforms for that 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] ipsec sa** command.

For more information about transform sets, refer to the *Cisco PIX Firewall and VPN Configuration Guide*.

show crypto ipsec commands

The **show crypto ipsec security-association lifetime** command displays the security-association lifetime value configured for a particular crypto map entry.

The **show crypto ipsec transform-set** [**tag** *transform-set-name*] command displays the configured transform sets.

The **show crypto ipsec sa** [**map** *map-name* | **address** | **identity**] [**detail**] command displays the settings used by current security associations.

Examples

The following example shortens the IPSec SA lifetimes. The time-out lifetime is shortened to 2700 seconds (45 minutes), and the traffic-volume lifetime is shortened to 2,304,000 kilobytes (10 megabytes per second for one half hour).

```
crypto ipsec security-association lifetime seconds 2700
crypto ipsec security-association lifetime kilobytes 2304000
```

The following is sample output from the **show crypto ipsec security-association lifetime** command:

```
show crypto ipsec security-association lifetime
Security-association lifetime: 4608000 kilobytes/120 seconds
```

The following configuration was in effect when the preceding **show crypto ipsec security-association lifetime** command was issued:

```
crypto ipsec security-association lifetime seconds 120
```

This example defines one transform set (named “standard”), which is used with an IPSec peer that supports the ESP protocol. Both an ESP encryption transform and an ESP authentication transform are specified in this example.

```
crypto ipsec transform-set standard esp-des esp-md5-hmac
```

The following is sample output for the **show crypto ipsec transform-set** command:

```
show crypto ipsec transform-set

Transform set combined-des-sha: {  esp-des esp-sha-hmac  }
   will negotiate = {  Tunnel,  },

Transform set combined-des-md5: {  esp-des esp-md5-hmac  }
   will negotiate = {  Tunnel,  },

Transform set t1: { esp-des esp-md5-hmac  }
   will negotiate = {  Tunnel,  },

Transform set t100: {  ah-sha-hmac  }
   will negotiate = {  Tunnel,  },

Transform set t2: {  ah-sha-hmac  }
   will negotiate = {  Tunnel,  },
```

```
   {  esp-des  }
   will negotiate = {  Tunnel,  },
```

The following configuration was in effect when the preceding **show crypto ipsec transform-set** command was issued:

```
crypto ipsec transform-set combined-des-sha esp-des esp-sha-hmac
crypto ipsec transform-set combined-des-md5 esp-des esp-md5-hmac
crypto ipsec transform-set t1 esp-des esp-md5-hmac
crypto ipsec transform-set t100 ah-sha-hmac
crypto ipsec transform-set t2 ah-sha-hmac esp-des
```

The following is sample output from the **show crypto ipsec sa** command:

```
show crypto ipsec sa

interface: outside
    Crypto map tag: firewall-robin, local addr. 172.21.114.123

   local  ident (addr/mask/prot/port): (172.21.114.123/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (172.21.114.67/255.255.255.255/0/0)
   current_peer: 172.21.114.67
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 10, #pkts encrypt: 10, #pkts digest 10
    #pkts decaps: 10, #pkts decrypt: 10, #pkts verify 10
    #send errors 10, #recv errors 0

     local crypto endpt.: 172.21.114.123, remote crypto endpt.: 172.21.114.67/500
     path mtu 1500, media mtu 1500
     current outbound spi: 20890A6F

     inbound esp sas:
      spi: 0x257A1039(628756537)
        transform: esp-des esp-md5-hmac ,
        in use settings ={Tunnel UDP-Encaps, }
        slot: 0, conn id: 26, crypto map: firewall-robin
        sa timing: remaining key lifetime (k/sec): (4607999/90)
        IV size: 8 bytes
        replay detection support: Y
     inbound ah sas:
     outbound esp sas:
      spi: 0x20890A6F(545852015)
        transform: esp-des esp-md5-hmac ,
        in use settings ={Tunnel,  }
        slot: 0, conn id: 27, crypto map: firewall-robin
        sa timing: remaining key lifetime (k/sec): (4607999/90)
        IV size: 8 bytes
        replay detection support: Y
     outbound ah sas:
interface: inside
    Crypto map tag: firewall-robin, local addr. 172.21.114.123

   local ident (addr/mask/prot/port): (172.21.114.123/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (172.21.114.67/255.255.255.255/0/0)
   current_peer: 172.21.114.67
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 10, #pkts encrypt: 10, #pkts digest 10
    #pkts decaps: 10, #pkts decrypt: 10, #pkts verify 10
    #send errors 10, #recv errors 0

     local crypto endpt.: 172.21.114.123, remote crypto endpt.: 172.21.114.67
     path mtu 1500, media mtu 1500
     current outbound spi: 20890A6F
        inbound esp sas:
```

```
 spi: 0x257A1039(628756537)
   transform: esp-des esp-md5-hmac  ,
   in use settings ={Tunnel,  }
   slot: 0, conn id: 26, crypto map: firewall-robin
   sa timing: remaining key lifetime (k/sec): (4607999/90)
   IV size: 8 bytes
   replay detection support: Y
inbound ah sas:
outbound esp sas:
 spi: 0x20890A6F(545852015)
   transform: esp-des esp-md5-hmac  ,
   in use settings ={Tunnel,  }
   slot: 0, conn id: 27, crypto map: firewall-robin
   sa timing: remaining key lifetime (k/sec): (4607999/90)
   IV size: 8 bytes
   replay detection support: Y
outbound ah sas:
```

crypto map

Create, modify, view or delete a crypto map entry. Also used to delete a crypto map set.

[no] **crypto map** *map-name* **client** [token] **authentication** *aaa-server-name* **[LOCAL]**

[no] **crypto map** *map-name* **client configuration address initiate | respond**

[no] **crypto map** *map-name* **interface** *interface-name*

[no] **crypto map** *map-name seq-num* **ipsec-isakmp | ipsec-manual** [**dynamic** *dynamic-map-name*]

[no] **crypto map** *map-name seq-num* **match address** *acl_name*

[no] **crypto map** *map-name seq-num* **set peer** {*ip_address* | *hostname*}

[no] **crypto map** *map-name seq-num* **set pfs** [**group1 | group2**]

[no] **crypto map** *map-name seq-num* **set security-association lifetime seconds** *seconds* | **kilobytes** *kilobytes*

[no] **crypto map** *map-name seq-num* **set session-key inbound | outbound ah** *spi hex-key-string*

[no] **crypto map** *map-name seq-num* **set session-key inbound | outbound esp** *spi* **cipher** *hex-key-string* [**authenticator** *hex-key-string*]

[no] **crypto map** *map-name seq-num* **set transform-set** *transform-set-name1* [*… transform-set-name6*]

show crypto map [**interface** *interface-name* | **tag** map-name]

Syntax Description

aaa-server-name	The name of the AAA server that will authenticate the user during IKE authentication. The AAA server options available are TACACS+, RADIUS, or LOCAL.
	If LOCAL is specified and the local user credential database is empty, the following warning message appears:
	`Warning:local database is empty! Use \Qusername' command to define local users.`
	Conversely, if the local database becomes empty when LOCAL is still present in the command, the following warning message appears:
	`Warning:Local user database is empty and there are still commands using LOCAL for authentication.`

acl_name	Identify the named encryption access list. This name should match the name argument of the named encryption access list being matched.
ah	Set the IPSec session key for the AH protocol. Specify **ah** when the crypto map entry's transform set includes an AH transform. AH protocol provides authentication via MD5-HMAC and SHA-HMAC.
authenticator	(Optional) Indicate that the key string is to be used with the ESP authentication transform. This argument is required only when the crypto map entry's transform set includes an ESP authentication transform.
cipher	Indicate that the key string to use with the ESP encryption transform.
dynamic	(Optional) Specify that this crypto map entry is to reference a pre-existing dynamic crypto map.
dynamic-map-name	(Optional) Specify the name of the dynamic crypto map set to be used as the policy template.
esp	Set the IPSec session key for the ESP protocol. Specify **esp** when the crypto map entry's transform set includes an ESP transform. ESP protocol provides both authentication and/or confidentiality. Authentication is done via MD5-HMAC, SHA-HMAC and NULL. Confidentiality is done via DES, 3DES, and NULL.
group1	Specify that IPSec should use the 768-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
group2	Specify that IPSec should use the 1024-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
hex-key-string	Specify the session key; enter in hexadecimal format. This is an arbitrary hexadecimal string of 16, 32, or 40 digits. If the crypto map's transform set includes the following: • DES algorithm, specify at least 16 hexadecimal digits per key. • MD5 algorithm, specify at least 32 hexadecimal digits per key. • SHA algorithm, specify 40 hexadecimal digits per key. Longer key sizes are simply hashed to the appropriate length.
hostname	Specify a peer by its IP address, or by its host name as defined by the PIX Firewall **name** command.
inbound	Set the inbound IPSec session key. (You must set both inbound and outbound keys.)
initiate	Indicate that the PIX Firewall will attempt to set IP addresses for each peer.

interface *interface-name*	Specify the identifying interface to be used by the PIX Firewall to identify itself to peers. If IKE is enabled, and you are using a certification authority (CA) to obtain certificates, this should be the interface with the address specified in the CA certificates.
ip_address	Specify a peer by its IP address.
ipsec-isakmp	Indicate that IKE will be used to establish the IPSec security associations for protecting the traffic specified by this crypto map entry.
ipsec-manual	Indicate that IKE will not be used to establish the IPSec security associations for protecting the traffic specified by this crypto map entry. **Note** Manual configuration of SAs is not supported on the PIX 501.
kilobytes *kilobytes*	Specify the volume of traffic (in kilobytes) that can pass between peers using a given security association before that security association expires. The default is 4,608,000 kilobytes.
map *map-name*	The name of the crypto map set.
match address	Specify an access list for a crypto map entry.
outbound	Set the outbound IPSec session key. (You must set both inbound and outbound keys.)
respond	Indicate that the PIX Firewall will accept requests for IP addresses from any requesting peer.
seconds *seconds*	Specify the number of seconds a security association will live before it expires. The default is 28,800 seconds (eight hours).
seq-num	The number you assign to the crypto map entry.
set peer	Specify an IPSec peer in a crypto map entry.
set pfs	Specify that IPSec should ask for perfect forward secrecy (PFS). With PFS, every time a new security association is negotiated, a new Diffie-Hellman exchange occurs. (This exchange requires additional processing time.)
set security-association lifetime	Set the lifetime a security association will last in either seconds or kilobytes. For use with either **seconds** or **kilobyte** keywords.
set session-key	Manually specify the IPSec session keys within a crypto map entry.
set transform-set	Specify which transform sets can be used with the crypto map entry.
spi	Specify the Security Parameter Index (SPI), a number that is used to uniquely identify a security association. The SPI is an arbitrary number you assign in the range of 256 to 4,294,967,295 (a hexidecimal value of FFFF FFFF). You can assign the same SPI to both directions and both protocols. However, not all peers have the same flexibility in SPI assignment. For a given destination address/protocol combination, unique SPI values must be used. The destination address is that of the PIX Firewall if inbound, the peer if outbound.
tag *map-name*	(Optional) Show the crypto map set with the specified map name.
token	Indicate a token-based server for user authentication is used.

transform1 *transform2* *transform3*	Specify up to three transforms. Transforms define the IPSec security protocol(s) and algorithm(s). Each transform represents an IPSec security protocol (ESP, AH, or both) plus the algorithm you want to use.
transform-set-name	The name of the transform set. For an ipsec-manual crypto map entry, you can specify only one transform set. For an ipsec-isakmp or dynamic crypto map entry, you can specify up to six transform sets.

Command Modes

Configuration mode.

Usage Guidelines

The sections that follow describe each **crypto map** command.

Note If a **crypto map** *map-name* **client configuration address initiate | respond** command configuration exists on the firewall, then the Cisco VPN Client version 3.x uses it.

crypto map client authentication

The **crypto map client authentication** command enables the Extended Authentication (Xauth) feature, which lets you prompt for a TACACS+, RADIUS, or LOCAL username and password during IKE authentication. You must first set up your AAA server configuration to use this feature, and be sure to specify the same AAA server name within the **crypto map client authentication** command statement as was specified in the **aaa-server** command statement.

This command tells the PIX Firewall during Phase 1 of IKE to use the Xauth (RADIUS, TACACS+, or LOCAL) challenge to authenticate IKE. If the Xauth fails, the IPSec security association will not be established, and the IKE security association will be deleted. Use the **no crypto map client authentication** command to restore the default value. The Xauth feature is not enabled by default.

Note Normally, when Xauth is enabled, an entry is added to the uauth table (as shown by the **show uauth/clear uauth** command) for the IP address assigned to the client. However, when using Xauth with the Easy VPN Remote feature in Network Extension Mode, the IPSEC tunnel is created from network to network, so the users behind the firewall cannot be associated with a single IP address. For this reason, a uauth entry cannot be created upon completion of Xauth. If AAA authorization or accounting services are required, you can enable the AAA authentication proxy to authenticate users behind the firewall. For more information on AAA authentication proxies, please refer to the **aaa** commands.

You cannot enable Xauth or IKE Mode Configuration on a interface when terminating an L2TP/IPSec tunnel using the Microsoft L2TP/IPSec client v1.0 (which is available on Windows NT, Windows XP, Windows 98 and Windows ME OS). Instead, you can do either of the following:

- Use a Windows 2000 L2TP/IPSec client, or
- Use the **isakmp key** *keystring* **address** *ip_address* **netmask** *mask* **no-xauth no-config-mode** command to exempt the L2TP client from Xauth and IKE Mode Configuration. However, if you exempt the L2TP client from Xauth or IKE Mode Configuration, all the L2TP clients must be grouped with the same ISAKMP pre-shared key or certificate and have the same fully qualified domain name.

The **crypto map client token authentication** command enables the PIX Firewall to interoperate with a Cisco VPN 3000 Client that is set up to use a token-based server for user authentication. The keyword **token** tells the PIX Firewall that the AAA server uses a token-card system and to prompt the user for username and password during IKE authentication. Use the **no crypto map client token authentication** command to restore the default value.

Note The remote user must be running one of the following:
Cisco VPN Client Version 3.x
Cisco VPN 3000 Client Version 2.5/2.6 or higher
Cisco Secure VPN Client Version 1.1 or higher

crypto map client configuration address

Use the **crypto map client configuration address** command to configure the IKE Mode Configuration on your PIX Firewall. IKE Mode Configuration allows the PIX Firewall to download an IP address to the remote peer (client) as part of an IKE negotiation. With the **crypto map client configuration address** command, you define the crypto map(s) that should attempt to configure the peer.

Use the **no crypto map client configuration address** command to restore the default value. IKE Mode Configuration is not enabled by default.

The keyword **initiate** indicates that the PIX Firewall will attempt to set IP addresses for each peer. The **respond** keyword indicates that the PIX Firewall will accept requests for IP addresses from any requesting peer.

Note If you use IKE Mode Configuration on the PIX Firewall, the routers handling the IPSec traffic must also support IKE Mode Configuration. Cisco IOS Release 12.0(6)T and higher supports the IKE Mode Configuration.

Refer to the *Cisco PIX Firewall and VPN Configuration Guide* for more information about IKE Mode Configuration.

The following examples show how to configure IKE Mode Configuration on your PIX Firewall:

```
crypto map mymap client configuration address initiate
crypto map mymap client configuration address respond
```

crypto map interface

The **crypto map interface** command applies a previously defined crypto map set to an interface. Use the **no crypto map interface** command to remove the crypto map set from the interface. Use the **show crypto map** [**interface** | **tag**] to view the crypto map configuration.

Use this command to assign a crypto map set to any active PIX Firewall interface. The PIX Firewall supports IPSec termination on any and all active interfaces. You must assign a crypto map set to an interface before that interface can provide IPSec services.

Only one crypto map set can be assigned to an interface. If multiple crypto map entries have the same *map-name* but a different *seq-num*, they are considered to be part of the same set and will all be applied to the interface. The crypto map entry with the lowest *seq-num* is considered the highest priority and will be evaluated first. A single crypto map set can contain a combination of ipsec-isakmp and ipsec-manual crypto map entries.

Note While a new crypto map instance is being added to the PIX Firewall, all clear and SSH traffic to the firewall interface stops because the crypto peer/ACL pair has not yet been defined. To workaround this, use PIX Device Manager (PDM) to add the new crypto map instance or, through the PIX Firewall CLI, remove the **crypto map interface** command from your configuration, add the new crypto map instance and fully configure the crypto peer/ACL pair, and then reapply the **crypto map interface** command back to the interface. In some conditions the CLI workaround is not acceptable as it temporarily stops VPN traffic also.

The use of the **crypto map interface** command re-initializes the security association database causing any currently established security associations to be deleted.

The following example assigns the crypto map set "mymap" to the outside interface. When traffic passes through the outside interface, the traffic will be evaluated against all the crypto map entries in the "mymap" set. When outbound traffic matches an access list in one of the "mymap" crypto map entries, a security association (if IPSec) will be established per that crypto map entry's configuration (if no security association or connection already exists).

```
crypto map mymap interface outside
```

The following is sample output from the **show crypto map** command:

```
show crypto map

Crypto Map: "firewall-robin" pif: outside local address: 172.21.114.123

Crypto Map "firewall-robin" 10 ipsec-isakmp
        Peer = 172.21.114.67
        access-list 141 permit ip host 172.21.114.123 host 172.21.114.67
        Current peer: 172.21.114.67
        Security-association lifetime: 4608000 kilobytes/120 seconds
        PFS (Y/N): N
        Transform sets={  t1,  }
```

The following configuration was in effect when the preceding **show crypto map** command was issued:

```
crypto map firewall-robin 10 ipsec-isakmp
crypto map firewall-robinrobin 10 set peer 172.21.114.67
crypto map firewall-robin 10 set transform-set t1
crypto map firewall-robin 10 match address 141
```

The following is sample output from the **show crypto map** command when manually established security associations are used:

```
show crypto map

Crypto Map "multi-peer" 20 ipsec-manual
        Peer = 172.21.114.67
        access-list 120 permit ip host 1.1.1.1 host 1.1.1.2
        Current peer: 172.21.114.67
        Transform sets={  t2,  }
        Inbound esp spi: 0,
         cipher key: ,
         auth_key: ,
        Inbound ah spi: 256,
            key: 0102030405060708090102030405060708090102030405060708090,
        Outbound esp spi: 0
         cipher key: ,
         auth key: ,
        Outbound ah spi: 256,
            key: 0102030405060708090102030405060708090102030405060708090,
```

The following configuration was in effect when the preceding **show crypto map** command was issued:

```
crypto map multi-peer 20 ipsec-manual
crypto map multi-peer 20 set peer 172.21.114.67
crypto map multi-peer 20 set session-key inbound ah 256
0102030405060708090102030405060708090102030405060708090
crypto map multi-peer 20 set session-key outbound ah 256
0102030405060708090102030405060708090102030405060708090
crypto map multi-peer 20 set transform-set t2
crypto map multi-peer 20 match address 120
```

crypto map ipsec-manual | ipsec-isakmp

To create or modify a crypto map entry, use the **crypto map ipsec-manual | ipsec-isakmp** command. To create or modify an ipsec-manual crypto map entry, use the **ipsec-manual option** of the command. To create or modify an ipsec-isakmp crypto map entry, use the **ipsec-isakmp** option of the command. Use the **no crypto map** command to delete a crypto map entry or set.

Note The **crypto map** command without a keyword creates an ipsec-isakmp entry by default.

After you define crypto map entries, you can use the **crypto map interface** command to assign the crypto map set to interfaces.

Crypto maps provide two functions: filtering/classifying traffic to be protected, and defining the policy to be applied to that traffic. The first use affects the flow of traffic on an interface; the second affects the negotiation performed (via IKE) on behalf of that traffic.

IPSec crypto maps link together definitions of the following:

- What traffic should be protected
- Which IPSec peer(s) the protected traffic can be forwarded to—these are the peers with which a security association can be established
- Which transform sets are acceptable for use with the protected traffic
- How keys and security associations should be used/managed (or what the keys are, if IKE is not used)

A crypto map set is a collection of crypto map entries each with a different seq-num but the same map-name. Therefore, for a given interface, you could have certain traffic forwarded to one peer with specified security applied to that traffic, and other traffic forwarded to the same or a different peer with different IPSec security applied. To accomplish this you would create two crypto map entries, each with the same map-name, but each with a different seq-num.

The number you assign to the seq-num argument should not be arbitrary. This number is used to rank multiple crypto map entries within a crypto map set. Within a crypto map set, a crypto map entry with a lower seq-num is evaluated before a map entry with a higher seq-num; that is, the map entry with the lower number has a higher priority.

Note Every static crypto map must define an access list and an IPsec peer. If either is missing, the crypto map is considered incomplete and any traffic that has not already been matched to an earlier, complete crypto map is dropped. Use the **show conf** command to ensure that every crypto map is complete. To fix an incomplete crypto map, remove the crypto map, add the missing entries, and reapply it.

The following example shows the minimum required crypto map configuration when IKE will be used to establish the security associations:

```
crypto map mymap 10 ipsec-isakmp
crypto map mymap 10 match address 101
crypto map mymap set transform-set my_t_set1
crypto map mymap set peer 10.0.0.1
```

The following example shows the minimum required crypto map configuration when the security associations are manually established:

```
crypto transform-set someset ah-md5-hmac esp-des
crypto map mymap 10 ipsec-manual
crypto map mymap 10 match address 102
crypto map mymap 10 set transform-set someset
crypto map mymap 10 set peer 10.0.0.5
crypto map mymap 10 set session-key inbound ah 256 9876543210987654987654321098765 4
crypto map mymap 10 set session-key outbound ah 256 fedcbafedcbafedcfedcbafedcbafedc
crypto map mymap 10 set session-key inbound esp 256 cipher 0123456789012345
crypto map mymap 10 set session-key outbound esp 256 cipher abcdefabcdefabcd
```

crypto map ipsec-isakmp dynamic

To specify that a given crypto map entry is to reference a pre-existing dynamic crypto map, use the **crypto map ipsec-isakmp dynamic** command.

Use the **crypto dynamic-map** command to create dynamic crypto map entries. After you create a dynamic crypto map set, use the **crypto map ipsec-isakmp dynamic** command to add the dynamic crypto map set to a static crypto map.

Give crypto map entries which reference dynamic map sets the lowest priority map entries so that inbound security association negotiation requests will try to match the static maps first. Only after the request does not match any of the static maps do you want it to be evaluated against the dynamic map set.

To make a crypto map entry that references a dynamic crypto map to be set to the lowest priority map entry, give the map entry the highest seq-num of all the map entries in a crypto map set.

The following example configures an IPSec crypto map set that includes a reference to a dynamic crypto map set.

Crypto map “mymap 10” allows security associations to be established between the PIX Firewall and either (or both) of two remote IPSec peers for traffic matching access list 101. Crypto map “mymap 20” allows either of two transform sets to be negotiated with the peer for traffic matching access list 102.

Crypto map entry “mymap 30” references the dynamic crypto map set “mydynamicmap,” which can be used to process inbound security association negotiation requests that do not match “mymap” entries 10 or 20. In this case, if the peer specifies a transform set that matches one of the transform sets specified in “mydynamicmap” for a flow “permitted” by the access list 103, IPSec will accept the request and set up security associations with the peer without previously knowing about the peer. If accepted, the resulting security associations (and temporary crypto map entry) are established according to the settings specified by the peer.

The access list associated with “mydynamicmap 10” is also used as a filter. Inbound packets that match a permit statement in this list are dropped for not being IPSec protected. (The same is true for access lists associated with static crypto maps entries.) Outbound packets that match a permit statement without an existing corresponding IPSec security association are also dropped.

The following example shows the configuration using “mydynamicmap”:

```
crypto map mymap 10 ipsec-isakmp
crypto map mymap 10 match address 101
crypto map mymap 10 set transform-set my_t_set1
crypto map mymap 10 set peer 10.0.0.1
crypto map mymap 10 set peer 10.0.0.2
crypto map mymap 20 ipsec-isakmp
crypto map mymap 10 match address 102
crypto map mymap 10 set transform-set my_t_set1 my_t_set2
crypto map mymap 10 set peer 10.0.0.3
crypto dynamic-map mydynamicmap 10
crypto dynamic-map mydynamicmap 10 match address 103
crypto dynamic-map mydynamicmap 10 set transform-set my_t_set1 my_t_set2 my_t_set3
crypto map mymap 30 ipsec-isakmp dynamic mydynamicmap
```

crypto map match address

To assign an access list to a crypto map entry, use the **crypto map match address** command. Use the **no crypto map match address** command to remove the access list from a crypto map entry.

This command is required for all static crypto map entries. If you are defining a dynamic crypto map entry (with the **crypto dynamic-map** command), this command is not required but is strongly recommended.

Use the **access-list** command to define this access list.

The access list specified with this command will be used by IPSec to determine which traffic should be protected by IPSec crypto and which traffic does not need protection. (Traffic that is permitted by the access list will be protected. Traffic that is denied by the access list will not be protected in the context of the corresponding crypto map entry.)

Note The crypto access list is not used to determine whether to permit or deny traffic through the interface. An access list applied directly to the interface with the **access-group** command makes that determination.

The crypto access list specified by this command is used when evaluating both inbound and outbound traffic. Outbound traffic is evaluated against the crypto access lists specified by the interface’s crypto map entries to determine if it should be protected by crypto, and if so (if traffic matches a permit entry), which crypto policy applies. (If necessary, in the case of static IPSec crypto maps, new security associations are established using the data flow identity as specified in the permit entry; in the case of dynamic crypto map entries, if no security association exists, the packet is dropped.) Inbound traffic is evaluated against the crypto access lists specified by the entries of the interface’s crypto map set to determine if it should be protected by crypto and, if so, which crypto policy applies. (In the case of IPSec, unprotected traffic is discarded because it should have been protected by IPSec.)

The access list is also used to identify the flow for which the IPSec security associations are established. In the outbound case, the permit entry is used as the data flow identity (in general). In the inbound case, the data flow identity specified by the peer must be “permitted” by the crypto access list.

The following example shows the minimum required crypto map configuration when IKE will be used to establish the security associations. (This example is for a static crypto map.)

```
crypto map mymap 10 ipsec-isakmp
crypto map mymap 10 match address 101
crypto map mymap 10 set transform-set my_t_set1
crypto map mymap 10 set peer 10.0.0.1
```

crypto map set peer

Use the crypto map set peer command to specify an IPSec peer in a crypto map entry. Use the **no crypto map set peer** command to remove an IPSec peer from a crypto map entry.

This command is required for all static crypto maps. If you are defining a dynamic crypto map (with the **crypto dynamic-map** command), this command is not required, and in most cases is not used because, in general, the peer is unknown.

For **ipsec-isakmp crypto map** entries, you can specify multiple peers by repeating this command. The peer that packets are actually sent to is determined by the last peer that the PIX Firewall 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.

For **ipsec-manual crypto** entries, you can specify only one peer per crypto map. If you want to change the peer, you must first delete the old peer and then specify the new peer.

The following example shows a crypto map configuration when IKE will be used to establish the security associations. In this example, a security association could be set up to either the peer at 10.0.0.1 or the peer at 10.0.0.2.

```
crypto map mymap 10 ipsec-isakmp
crypto map mymap 10 match address 101
crypto map mymap 10 set transform-set my_t_set1
crypto map mymap 10 set peer 10.0.0.1 10.0.0.2
```

crypto map set pfs

The crypto map set pfs command sets IPSec to ask for perfect forward secrecy (PFS) when requesting new security associations for this crypto map entry, or that IPSec requires PFS when receiving requests for new security associations. To specify that IPSec should not request PFS, use the **no crypto map set pfs** command. This command is only available for ipsec-isakmp crypto map entries and dynamic crypto map entries.

By default, PFS is not requested.

With PFS, every time a new security association is negotiated, a new Diffie-Hellman exchange occurs, which requires additional processing time. PFS adds another level of security because if one key is ever cracked by an attacker, only the data sent with that key will be compromised.

During negotiation, this command causes IPSec to request PFS when requesting new security associations for the crypto map entry. The default (group1) is sent if the **set pfs** statement does not specify a group.

If the peer initiates the negotiation and the local configuration specifies PFS, the peer must perform a PFS exchange or the negotiation will fail. If the local configuration does not specify a group, a default of group1 will be assumed, and an offer of either group1 or group2 will be accepted. If the local configuration specifies group2, that group must be part of the peer's offer or the negotiation will fail. If the local configuration does not specify PFS, it will accept any offer of PFS from the peer.

The 1024-bit Diffie-Hellman prime modulus group, group2, provides more security than group1, but requires more processing time than group1.

Note IKE negotiations with a remote peer may hang when a PIX Firewall has numerous tunnels that originate from the PIX Firewall and terminate on a single remote peer. This problem occurs when PFS is not enabled, and the local peer requests many simultaneous rekey requests. If this problem occurs, the IKE security association will not recover until it has timed out or until you manually clear it with the **clear [crypto] isakmp sa** command. PIX Firewall units configured with many tunnels to many peers or many clients sharing the same tunnel are not affected by this problem. If your configuration is affected, enable PFS with the **crypto map** *mapname seqnum* **set pfs** command.

The following example specifies that PFS should be used whenever a new security association is negotiated for the crypto map “mymap 10”:

```
crypto map mymap 10 ipsec-isakmp
crypto map mymap 10 set pfs group2
```

crypto map set security-association lifetime

To override (for a particular crypto map entry) the global lifetime value, which is used when negotiating IPSec security associations, use the **crypto map set security-association lifetime** command. To reset a crypto map entry's lifetime value to the global value, use the **no crypto map set security-association lifetime** command.

The crypto map's security associations are negotiated according to the global lifetimes.

This command is only available for ipsec-isakmp crypto map entries and dynamic crypto map entries.

IPSec security associations use shared secret keys. These keys and their security associations time out together.

Assuming that the particular crypto map entry has lifetime values configured, when the PIX Firewall requests new security associations during security association negotiation, it will specify its crypto map lifetime value in the request to the peer; it will use this value as the lifetime of the new security associations. When the PIX Firewall receives a negotiation request from the peer, it will use the smaller of the lifetime value proposed by the peer or the locally configured lifetime value as the lifetime of the new security associations.

There are two lifetimes: a “timed” lifetime and a “traffic-volume” lifetime. The session keys/security association expires after the first of these lifetimes is reached.

If you change a lifetime, the change will not be applied to existing security associations, but will be used in subsequent negotiations to establish security associations for data flows supported by this crypto map entry. 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] ipsec sa** command. See the **clear [crypto] ipsec sa** command for more details.

To change the timed lifetime, use the **crypto map set security-association lifetime seconds** command. The timed lifetime causes the keys and security association to time out after the specified number of seconds have passed.

To change the traffic-volume lifetime, use the **crypto map set security-association lifetime kilobytes** command. The traffic-volume lifetime causes the key and security association to time out after the specified amount of traffic (in kilobytes) has been protected by the security association's key.

Shorter lifetimes can make it harder to mount a successful key recovery attack, because the attacker has less data encrypted under the same key to work with.

However, shorter lifetimes require more CPU processing time.

The lifetime values are ignored for manually established security associations (security associations installed via an ipsec-manual crypto map entry).

The following example shortens the timed lifetime for a particular crypto map entry, because there is a higher risk that the keys could be compromised for security associations belonging to the crypto map entry. The traffic-volume lifetime is not changed because there is not a high volume of traffic anticipated for these security associations. The timed lifetime is shortened to 2700 seconds (45 minutes).

```
crypto map mymap 10 ipsec-isakmp
set security-association lifetime seconds 2700
```

crypto map set session-key

To manually specify the IPSec session keys within a crypto map entry, use the **crypto map set session-key** command. Use the **no crypto map set session-key** command to remove IPSec session keys from a crypto map entry. This command is only available for ipsec-manual crypto map entries.

If the crypto map's transform set includes an AH protocol, you must define IPSec keys for AH for both inbound and outbound traffic. If the crypto map's transform set includes an ESP encryption protocol, you must define IPSec keys for ESP encryption for both inbound and outbound traffic. If the crypto map's transform set includes an ESP authentication protocol, you must define IPSec keys for ESP authentication for inbound and outbound traffic.

When you define multiple IPSec session keys within a single crypto map, you can assign the same Security Parameter Index (SPI) number to all the keys. The SPI is used to identify the security association used with the crypto map. However, not all peers have the same flexibility in SPI assignment.

You may have to coordinate SPI assignment with the peer's network administrator, making certain that the same SPI is not used more than once for the same destination address/protocol combination.

Security associations established using this command do not expire (unlike security associations established using IKE).

The PIX Firewall unit's session keys must match its peer's session keys.

If you change a session key, the security association using the key will be deleted and reinitialized.

The following example shows a crypto map entry for manually established security associations. The transform set "t_set" includes only an AH protocol.

```
crypto ipsec transform-set t_set ah-sha-hmac
crypto map mymap 20 ipsec-manual
crypto map mymap 20 match address 102
crypto map mymap 20 set transform-set t_set
crypto map mymap 20 set peer 10.0.0.21
crypto map mymap 20 set session-key inbound ah 300
1111111111111111111111111111111111111111
crypto map mymap 20 set session-key outbound ah 300
22222222222222222222222222222222222222
```

The following example shows a crypto map entry for manually established security associations. The transform set "someset" includes both an AH and an ESP protocol, so session keys are configured for both AH and ESP for both inbound and outbound traffic. The transform set includes both encryption and authentication ESP transforms, so session keys are created for both using the **cipher** and **authenticator** keywords.

```
crypto ipsec transform-set someset ah-sha-hmac esp-des esp-sha-hmac
crypto map mymap 10 ipsec-manual
crypto map mymap 10 match address 101
crypto map mymap 10 set transform-set someset
crypto map mymap 10 set peer 10.0.0.1
crypto map mymap 10 set session-key inbound ah 300
9876543210987654321098765432109876543210
crypto map mymap 10 set session-key outbound ah 300
fedcbafedcbafedcbafedcbafedcbafedcbafedc
crypto map mymap 10 set session-key inbound esp 300 cipher 0123456789012345
    authenticator 00001111222233334444555566667777888899999
crypto map mymap 10 set session-key outbound esp 300 cipher abcdefabcdefabcd
    authenticator 99998888777766665555444433332222111110000
```

crypto map set transform-set

To specify which transform sets can be used with the crypto map entry, use the **crypto map set transform-set** command. Use the **no crypto map set transform-set** command to remove all transform sets from a crypto map entry.

This command is required for all static and dynamic crypto map entries.

For an **ipsec-isakmp crypto map** entry, you can list up to six transform sets with this command. List the higher priority transform sets first.

If the local PIX Firewall initiates the negotiation, the transform sets are presented to the peer in the order specified in the **crypto map** command statement. If the peer initiates the negotiation, the local PIX Firewall accepts the first transform set that matches one of the transform sets specified in the crypto map entry.

The first matching transform set that is found at both peers is used for the security association. If no match is found, IPSec will not establish a security association. The traffic will be dropped because there is no security association to protect the traffic.

For an **ipsec-manual crypto map** command statement, you can specify only one transform set. If the transform set does not match the transform set at the remote peer's crypto map, the two peers will fail to correctly communicate because the peers are using different rules to process the traffic.

If you want to change the list of transform sets, respecify the new list of transform sets to replace the old list. This change is only applied to **crypto map** command statements that reference this 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] ipsec sa** command.

Any transform sets included in a **crypto map** command statement must previously have been defined using the **crypto ipsec transform-set** command.

Examples

The following example shows how the **crypto map client authentication** command is used. This example sets up the IPSec rules for VPN encryption IPSec. The **ip**, **nat**, **aaa-server** command statements establish the context for the IPSec-related commands.

```
ip address inside 10.0.0.1 255.255.255.0
ip address outside 168.20.1.5 255.255.255.0
dealer 10.1.2.1-10.1.2.254
nat (inside) 0 access-list 80
aaa-server TACACS+ protocol tacacs+
aaa-server TACACS+ (inside) host 10.0.0.2 secret123
crypto ipsec transform-set pc esp-des esp-md5-hmac
crypto dynamic-map cisco 4 set transform-set pc
crypto map partner-map 20 ipsec-isakmp dynamic cisco
crypto map partner-map client configuration address initiate
crypto map partner-map client authentication TACACS+
crypto map partner-map interface outside
isakmp key cisco1234 address 0.0.0.0 netmask 0.0.0.0
isakmp client configuration address-pool local dealer outside
isakmp policy 8 authentication pre-share
isakmp policy 8 encryption des
isakmp policy 8 hash md5
isakmp policy 8 group 1
isakmp policy 8 lifetime 86400
```

The following example shows how the **crypto map client token authentication** command is used. This example sets up the IPSec rules for VPN encryption IPSec. The **ip**, **nat**, **aaa-server** command statements establish the context for the IPSec-related commands.

```
ip address inside 10.0.0.1 255.255.255.0
ip address outside 168.20.1.5 255.255.255.0
ip local pool dealer 10.1.2.1-10.1.2.254
nat (inside) 0 access-list 80
aaa-server RADIUS protocol radius
aaa-server RADIUS (inside) host 10.0.0.2 secret123
crypto ipsec transform-set pc esp-des esp-md5-hmac
crypto dynamic-map cisco 4 set transform-set pc
crypto map partner-map 20 ipsec-isakmp dynamic cisco
crypto map partner-map client configuration address initiate
crypto map partner-map client token authentication RADIUS
crypto map partner-map interface outside
isakmp key cisco1234 address 0.0.0.0 netmask 0.0.0.0
isakmp client configuration address-pool local dealer outside
isakmp policy 8 authentication pre-share
isakmp policy 8 encryption des
isakmp policy 8 hash md5
isakmp policy 8 group 1
isakmp policy 8 lifetime 86400
```

The following example defines two transform sets and specifies that they can both be used within a crypto map entry. (This example applies only when IKE is used to establish security associations. With crypto maps used for manually established security associations, only one transform set can be included in a given **crypto map** command statement.)

```
crypto ipsec transform-set my_t_set1 esp-des esp-sha-hmac
crypto ipsec transform-set my_t_set2 ah-sha-hmac esp-des esp-sha-hmac
crypto map mymap 10 ipsec-isakmp
crypto map mymap 10 match address 101
crypto map mymap 10 set transform-set my_t_set1 my_t_set2
crypto map mymap set peer 10.0.0.1 10.0.0.2
```

In this example, when traffic matches access list 101 the security association can use either transform set “my_t_set1” (first priority) or “my_t_set2” (second priority), depending on which transform set matches the remote peer's transform sets.

CHAPTER 5

D through F Commands

debug

You can debug packets or ICMP tracings through the PIX Firewall. The **debug** command provides information that helps troubleshoot protocols operating with and through the PIX Firewall.

[**no**] **debug aaa** [**authentication | authorization| accounting | internal**]

[**no**] **debug access-list all | standard | turbo**

[**no**] **debug arp**

[**no**] **debug crypto ca** [*level*]

[**no**] **debug ctiqbe**

[**no**] **debug crypto ipsec** [*level*]

[**no**] **debug crypto isakmp** [*level*]

[**no**] **debug crypto vpnclient**

[**no**] **debug dhcpc detail | error | packet**

[**no**] **debug dhcpd event | packet**

[**no**] **debug dhcprelay event | packet | error**

[**no**] **debug dns** {**resolver | all**}

[**no**] **debug fixup** {**udp | tcp**}

[**no**] **debug fover** *option*

[**no**] **debug h323 h225** [**asn | event**]

[**no**] **debug h323 h245** [**asn | event**]

[**no**] **debug h323 ras** [**asn | event**]

[**no**] **debug icmp trace**

[**no**] **debug ils**

[**no**] **debug ospf** [**adj | database-timer | events |f lood | lsa-generation | packet | tree | retransmission | spf** [**external | internal |intra**]]

[**no**] **debug mgcp** [**messages | parser | sessions**]

[**no**] **debug ntp** [**adjust | authentication | events | loopfilter | packets | params | select | sync | validity**]

[**no**] **debug packet** *if_name* [**src** *source_ip* [**netmask** *mask*]] [**dst** *dest_ip* [**netmask** *mask*]] [[**proto icmp**] | [**proto tcp** [**sport** *src_port*] [**dport** *dest_port*]] | [**proto udp** [**sport** *src_port*] [**dport** *dest_port*]] [**rx | tx | both**]

[**no**] **debug pdm history**

[**no**] **debug ppp error | io | uauth | upap | chap | negotiation**

[**no**] **debug pppoe event | error | packet**

[**no**] **deubg pptp**

[**no**] **debug radius** [**session | all | user** *username*]

[**no**] **debug rip**

[**no**] **debug route**

[**no**] **debug rtsp**

[**no**] **debug sip**

[**no**] **debug skinny**

[**no**] **debug sqlnet**

[**no**] **debug ssh**

[**no**] **debug ssl** [**cypher | device**]

[**no**] **debug vpdn event | error | packet**

[no] debug xdmcp

no debug all

undebug all

show debug

Syntax Description

aaa	Displays authentication, authorization, and accounting information.
access-list	Displays access list configuration information.
adjust	Displays NTP clock adjustments.
all	Displays both standard and TurboACL access list information.
authentication	Displays NTP clock authentication.

both	Displays both received and transmitted packets.
chap	Displays CHAP/MS-CHAP authentication.
crypto ca	Displays information about certification authority (CA) traffic.
crypto ipsec	Displays information about IPSec traffic.
crypto isakmp	Displays information about IKE traffic.
crypto vpnclient	Displays information about the firewall EasyVPN client.
ctiqbe	Displays information about CTI Quick Buffer Encoding (CTIQBE), which is used with Cisco TAPI/JTAPI applications.
cypher	Display information about the cipher negotiation between the HTTP server and the client.
device	Displays information about the SSL device including session initiation and ongoing status.
dhcpc detail	Displays detailed information about the DHCP client packets.
dhcpc error	Displays error messages associated with the DHCP client.
dhcpc packet	Displays packet information associated with the DHCP client.
dhcpd event	Displays event information associated with the DHCP server.
dhcpd packet	Displays packet information associated with the DHCP server.
dhcprelay	Displays DHCP Relay Agent information.
dns {**resolver** | **all**}	Displays DNS debugging information. The **resolver** option collects DNS resolution information, and the **all** option collects all DNS information.
dport *dest_port*	Destination port.
dst *dest_ip*	Destination IP address.
events	Displays NTP event information.
fixup {udp | tcp}	Displays fixup information, using either UDP or TCP.
fover *option*	Displays failover information. Refer to Table 5-1 for the *options*.
h225 asn	Displays the output of the decoded PDUs.
h225 events	Displays the events of the H.225 signaling, or turn both traces on.
h245 asn	Displays the output of the decoded PDUs.
h245 events	Displays the events of the H.245 signaling, or turn both traces on.
h323	Displays information about the packet-based multimedia communications systems standard.
icmp	Displays information about ICMP traffic.
if_name	Interface name from which the packets are arriving; for example, to monitor packets coming into the PIX Firewall from the outside, set *if_name* to **outside**.
ils	Displays Internet Locator Service (ILS) fixup information (used in LDAP services).

level	The level of debugging feedback. The higher the level number, the more information is displayed. The default *level* is 1. The levels correspond to the following events: • Level 1: Interesting events • Level 2: Normative and interesting events • Level 3: Diminutive, normative, and interesting events Refer to the "Examples" section at the end of this command page for an example of how the debugging level appears within the **show debug** command.
loopfilter	Displays NTP loop filter information.
messages	Displays debug information for MGCP messages.
negotiation	Equivalent of the **error, uauth, upap and chap debug** command options.
netmask *mask*	Network mask.
packet	Displays packet information.
packets	Displays NTP packet information.
params	Displays NTP clock parameters.
parser	Displays debug information about parsing MGCP messages.
pdm history	Turns on the PDM history metrics debugging information. The **no** version of this command disables PDM history metrics debugging.
ppp	Debugs L2TP or PPTP traffic, which is configured with the **vpdn** command.
ppp error	Displays L2TP or PPTP PPP virtual interface error messages.
ppp io	Display the packet information for L2TP or PPTP PPP virtual interface.
ppp uauth	Displays the L2TP or PPTP PPP virtual interface AAA user authentication debugging messages.
pppoe error	Displays PPPoE error messages.
pppoe event	Displays PPPoE event information.
pppoe packet	Displays PPPoE packet information.
pptp	Displays PPTP traffic information.
proto icmp	Displays ICMP packets only.
proto tcp	Displays TCP packets only.
proto udp	Displays UDP packets only.
radius all	Enables all RADIUS debug options.
radius session	Logs RADIUS session information and the attributes of sent and received RADIUS packets.
ras asn	Displays the output of the decoded PDUs.
ras events	Displays the events of the RAS signaling, or turn both traces on.
route	Displays information from the PIX Firewall routing module.
rx	Displays only packets received at the PIX Firewall.
select	Displays NTP clock selections.
sessions	Displays debug information for MGCP sessions.
sip	Debug the fixup Session Initiation Protocol (SIP) module.
skinny	Debugs SCCP protocol activity. (Using this option is system-resources intensive and may impact performance on high traffic network segments.)

sport *src_port*	Source port. See the "Ports" section in "Chapter 2, "Using PIX Firewall Commands" for a list of valid port literal names.
sqlnet	Debugs SQL*Net traffic.
src *source_ip*	Source IP address.
ssh	Debug information and error messages associated with the **ssh** command.
ssl	Debug information and error messages associated with the **ssl** command.
standard	Displays non-TurboACL access list information.
sync	Displays NTP clock synchronization.
turbo	Displays TurboACL access list information.
tx	Displays only packets that were transmitted from the PIX Firewall.
upap	Displays PAP authentication.
user *username*	Specifies to display information for an individual *username* only.
validity	Displays NTP peer clock validity.
vpdn error	Display L2TP or PPTP protocol error messages.
vpdn event	Display L2TP or PPTP tunnel event change information.
vpdn packet	Display L2TP or PPTP packet information about PPTP traffic.
xdmcp	Display information about the xdmcp negotiation

Defaults

MGCP debugging is disabled by default.

Command Modes

Configuration mode unless otherwise specified.

The **debug mgcp** command is available in privileged mode.

Usage Guidelines

The **debug** command lets you view debug information. The **show debug** command displays the current state of tracing. You can debug the contents of network layer protocol packets with the **debug packet** command.

Note Use of the **debug** commands may slow down traffic on busy networks.

Use of the **debug packet** command on a PIX Firewall experiencing a heavy load may result in the output displaying so fast that it may be impossible to stop the output by entering the **no debug packet** command from the console. You can enter the **no debug packet** command from a Telnet session.

To let users ping through the PIX Firewall, add the **access-list** *acl_grp* **permit icmp any any** command statement to the configuration and bind it to each interface you want to test with the **access-group** command. This lets pings go outbound and inbound.

To stop a **debug packet trace** command, enter the following command:

```
no debug packet if_name
```

Replace *if_name* with the name of the interface; for example, **inside**, **outside**, or a perimeter interface name.

no debug all and undebug all

The **no debug all** and **undebug all** commands stop any and all debug messages from being displayed.

debug crypto

When creating your digital certificates, use the **debug crypto ca** command to ensure that the certificate is created correctly. Important error messages only display when the **debug crypto ca** command is enabled. For example, if you enter an Entrust fingerprint value incorrectly, the only warning message that indicates the value is incorrect appears in the **debug crypto ca** command output.

Output from the **debug crypto ipsec** and **debug crypto isakmp** commands does not display in a Telnet console session.

debug dhcpc

The **debug dhcpc detail** command displays detailed packet information about the DHCP client. The **debug dhcpc error** command displays DHCP client error messages. The **debug dhcpc packet** command displays packet information about the DHCP client. Use the **no** form of the **debug dhcpc** command to disable debugging.

The **debug dhcpd event** command displays event information about the DHCP server. The **debug dhcpd packet** command displays packet information about the DHCP server. Use the **no** form of the **debug dhcpd** commands to disable debugging.

debug h323

The **debug h323** command lets you debug H.323 connections. Use the **no** form of the command to disable debugging. This command works when the **fixup protocol h323** command is enabled.

Note The **debug h323** command, particularly the **debug h323 h225 asn**, **debug h323 h245 asn**, and **debug h323 ras asn** commands, might delay the sending of messages and cause slower performance in a real-time environment.

debug icmp

The **debug icmp trace** command shows ICMP packet information, the source IP address, and the destination address of packets arriving, departing, and traversing the PIX Firewall including pings to the PIX Firewall unit's own interfaces.

To stop a **debug icmp trace** command, enter the following command:

```
no debug icmp trace
```

debug mgcp

The **debug mgcp** command displays debug information for Media Gateway Control Protocol (MGCP) traffic. Without any options explicitly specified, the **debug mgcp** command enables all three MGCP debug options. The **no debug mgcp** command, without any options explicitly specified, disables all MGCP debugging.

debug ospf

The **debug ospf** command enables all OSPF debugging options, and the **no debug ospf** command disables all OSPF debugging options.

The **debug ospf spf** command enables all SPF options, and the **no debug ospf spf** command disables all SPF options.

debug sqlnet

The **debug sqlnet** command reports on traffic between Oracle SQL*Net clients and servers through the PIX Firewall.

debug ssh

The **debug ssh** command reports on information and error messages associated with the **ssh** command.

debug pptp

The **debug pptp** and **debug vpdn** commands provide information about PPTP traffic. PPTP is configured with the **vpdn** command.

debug fover

Table 5-1 lists the options for the **debug fover** command.

Table 5-1 debug fover Command Options

Option	Description
cable	Failover cable status
fail	Failover internal exception
fmsg	Failover message
get	IP network packet received
ifc	Network interface status trace
lanrx	LAN-based failover receive process messages
lanretx	LAN-based failover retransmit process messages
lantx	LAN-based failover transmit process messages
lancmd	LAN-based failover main thread messages
open	Failover device open
put	IP network packet transmitted
rx	Failover cable receive
rxdmp	Cable recv message dump (serial console only)
rxip	IP network failover packet received
tx	Failover cable transmit
txdmp	Cable xmit message dump (serial console only)
txip	IP network failover packet transmit
verify	Failover message verify
switch	Failover Switching status

Trace Channel Feature

The **debug packet** command sends its output to the Trace Channel. All other **debug** commands do not. Use of Trace Channel changes the way you can view output on your screen during a PIX Firewall console or Telnet session.

If a **debug** command does not use Trace Channel, each session operates independently, which means any commands started in the session only appear in the session. By default, a session not using Trace Channel has output disabled by default.

The location of the Trace Channel depends on whether you have a simultaneous Telnet console session running at the same time as the console session, or if you are using only the PIX Firewall serial console:

- If you are only using the PIX Firewall serial console, all **debug** commands display on the serial console.
- If you have both a serial console session and a Telnet console session accessing the console, then no matter where you enter the **debug** commands, the output displays on the Telnet console session.
- If you have two or more Telnet console sessions, the first session is the Trace Channel. If that session closes, the serial console session becomes the Trace Channel. The next Telnet console session that accesses the console will then become the Trace Channel.

The **debug** commands, except the debug crypto commands, are shared between all Telnet and serial console sessions.

Note The downside of the Trace Channel feature is that if one administrator is using the serial console and another administrator starts a Telnet console session, the serial console **debug** command output will suddenly stop without warning. In addition, the administrator on the Telnet console session will suddenly be viewing **debug** command output, which may be unexpected. If you are using the serial console and **debug** command output is not appearing, use the **who** command to see if a Telnet console session is running.

Examples

The following is partial sample output from the **debug dhcpc packet** and the **debug dhcpc detail** commands. The **ip address dhcp setroute** command was configured after entering the **debug dhcpc** commands to obtain debugging information.

```
debug dhcpc packet
debug dhcpc detail
ip address outside dhcp setroute
DHCP:allocate request
DHCP:new entry. add to queue
DHCP:new ip lease str = 0x80ce8a28
DHCP:SDiscover attempt # 1 for entry:
Temp IP addr:0.0.0.0 for peer on Interface:outside
Temp sub net mask:0.0.0.0
   DHCP Lease server:0.0.0.0, state:1 Selecting
   DHCP transaction id:0x8931
   Lease:0 secs, Renewal:0 secs, Rebind:0 secs
   Next timer fires after:2 seconds
   Retry count:1   Client-ID:cisco-0000.0000.0000-outside

DHCP:SDiscover:sending 265 byte length DHCP packet
DHCP:SDiscover 265 bytes
DHCP Broadcast to 255.255.255.255 from 0.0.0.0
DHCP client msg received, fip=10.3.2.2, fport=67
DHCP:Received a BOOTREP pkt
DHCP:Scan:Message type:DHCP Offer
DHCP:Scan:Server ID Option:10.1.1.69 = 450A44AB
    DHCP:Scan:Server ID Option:10.1.1.69 = 450A44AB
DHCP:Scan:Lease Time:259200
DHCP:Scan:Subnet Address Option:255.255.254.0
DHCP:Scan:DNS Name Server Option:10.1.1.70, 10.1.1.140
DHCP:Scan:Domain Name:example.com
```

```
DHCP:Scan:NBNS Name Server Option:10.1.2.228, 10.1.2.87
DHCP:Scan:Router Address Option:10.3.2.1
DHCP:rcvd pkt source:10.3.2.2, destination: 255.255.255.255
...
```

The following example executes the **debug icmp trace** command:

```
debug icmp trace
```

When you ping a host through the PIX Firewall from any interface, trace output displays on the console. The following example shows a successful ping from an external host (209.165.201.2) to the PIX Firewall unit's outside interface (209.165.201.1).

```
Inbound ICMP echo reply (len 32 id 1 seq 256) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 512) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 512) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 768) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 768) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 1024) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 1024) 209.165.201.1 > 209.165.201.2
NO DEBUG ICMP TRACE
ICMP trace off
```

This example shows that the ICMP packet length is 32 bytes, the ICMP packet identifier is 1, and the ICMP sequence number. The ICMP sequence number starts at 0 and is incremented each time a request is sent.

The following is sample output from the **show debug** command output:

```
show debug
debug ppp error
debug vpdn event
debug crypto ipsec 1
debug crypto isakmp 1
debug crypto ca 1
debug icmp trace
debug packet outside both
debug sqlnet
```

The preceding sample output includes the **debug crypto** commands.

The following example shows debugging messages for Unity client negotiation using Diffie-Hellman group 5:

```
pixfirewall(config)# debug crypto isakmp

check_isakmp_proposal:
is_auth_policy_configured: auth 1
is_auth_policy_configured: auth 4
ISAKMP (0): Checking ISAKMP transform 1 against priority 8 policy
ISAKMP:      encryption 3DES-CBC
ISAKMP:      hash SHA
ISAKMP:      default group 5
ISAKMP:      extended auth RSA sig
ISAKMP:      life type in seconds
ISAKMP:      life duration (VPI) of  0x0 0x20 0xc4 0x9b
ISAKMP (0): atts are not acceptable. Next payload is 3
ISAKMP (0): Checking ISAKMP transform 2 against priority 8 policy
ISAKMP:      encryption 3DES-CBC
ISAKMP:      hash MD5
ISAKMP:      default group 5
ISAKMP:      extended auth RSA sig
ISAKMP:      life type in seconds
ISAKMP:      life duration (VPI) of  0x0 0x20 0xc4 0x9b
```

```
ISAKMP (0): atts are not acceptable. Next payload is 3
ISAKMP (0): Checking ISAKMP transform 3 against priority 8 policy
ISAKMP:      encryption 3DES-CBC
ISAKMP:      hash SHA
ISAKMP:      default group 5
ISAKMP:      auth RSA sig
ISAKMP:      life type in seconds
ISAKMP:      life duration (VPI) of  0x0 0x20 0xc4 0x9b
ISAKMP (0): atts are not acceptable. Next payload is 3
ISAKMP (0): Checking ISAKMP transform 4 against priority 8 policy
ISAKMP:      encryption 3DES-CBC
ISAKMP:      hash MD5
ISAKMP:      default group 5
ISAKMP:      auth RSA sig
ISAKMP:      life type in seconds
ISAKMP:      life duration (VPI) of  0x0 0x20 0xc4 0x9b
```

ISAKMP (0): atts are acceptable. Next payload is 3

The following example shows possible output for the **debug mgcp messages** command:

```
17: MGCP: Retransmitted command RSIP
        Gateway IP      gate-1
        Transaction ID  1
18: MGCP: Expired command RSIP
        Gateway IP      gate-1
        Transaction ID  1
19: MGCP: New command RSIP
        Gateway IP      gate-1
        Transaction ID  1
        Endpoint name   d001
        Call ID
        Connection ID
        Media IP        0.0.0.0
        Media port      0
        Flags           0x80
20: MGCP: Retransmitted command RSIP
        Gateway IP      gate-1
        Transaction ID  1
```

The following example shows possible output for the **debug mgcp parser** command:

```
28: MGCP packet:
RSIP 1 d001@10.10.10.11 MGCP 1.0
RM: restart

29: MGCP: command verb - RSIP
30: MGCP: transaction ID - 1
31: MGCP: endpoint name - d001
32: MGCP: header parsing succeeded
33: MGCP: restart method - restart
34: MGCP: payload parsing succeeded
35: MGCP packet:
RSIP 1 d001@10.10.10.11 MGCP 1.0
RM: restart

36: MGCP: command verb - RSIP
37: MGCP: transaction ID - 1
38: MGCP: endpoint name - d001
39: MGCP: header parsing succeeded
40: MGCP: restart method - restart
41: MGCP: payload parsing succeeded
```

The following example shows possible output for the **debug mgcp sessions** command:

```
91: NAT::requesting UDP conn for generic-pc-2/6166 [209.165.202.128/0]
        from dmz/ca:generic-pc-2/2427 to outside:generic-pc-1/2727
92: NAT::reverse route: embedded host at dmz/ca:generic-pc-2/6166
93: NAT::table route: embedded host at outside:209.165.202.128/0
94: NAT::pre-allocate connection for outside:209.165.202.128 to dmz/ca:generic-pc-2/6166
95: NAT::found inside xlate from dmz/ca:generic-pc-2/0 to outside:209.165.201.15/0
96: NAT::outside NAT not needed
97: NAT::created UDP conn dmz/ca:generic-pc-2/6166 <-> outside:209.165.202.128/0
98: NAT::created RTCP conn dmz/ca:generic-pc-2/6167 <-> outside:209.165.202.128/0
99: NAT::requesting UDP conn for 209.165.202.128/6058 [generic-pc-2/0]
        from dmz/ca:genericgeneric-pc-2/2427 to outside:generic-pc-1/2727
100: NAT::table route: embedded host at outside:209.165.202.128/6058
101: NAT::reverse route: embedded host at dmz/ca:generic-pc-2/0
102: NAT::pre-allocate connection for dmz/ca:generic-pc-2 to outside:209.165.202.128/6058
103: NAT::found inside xlate from dmz/ca:generic-pc-2/0 to outside:209.165.201.15/0
104: NAT::outside NAT not needed
105: NAT::created UDP conn dmz/ca:generic-pc-2/0 <-> outside:209.165.202.128/6058
106: NAT::created RTCP conn dmz/ca:generic-pc-2/0 <-> outside:209.165.202.128/6059
107: MGCP: New session
        Gateway IP     generic-pc-2
        Call ID        9876543210abcdef
        Connection ID  6789af54c9
        Endpoint name  aaln/1
        Media lcl port 6166
        Media rmt IP   209.165.202.128
        Media rmt port 6058
108: MGCP: Expired session, active 0:06:05
        Gateway IP      generic-pc-2
        Call ID         9876543210abcdef
        Connection ID   6789af54c9
        Endpoint name   aaln/1
        Media lcl port  6166
        Media rmt IP    209.165.202.128
        Media rmt port  6058
```

You can debug the contents of packets with the **debug packet** command:

```
debug packet inside
--------- PACKET ---------
-- IP --
4.3.2.1 ==>     255.3.2.1
        ver = 0x4       hlen = 0x5      tos = 0x0       tlen = 0x60
        id = 0x3902     flags = 0x0     frag off=0x0
        ttl = 0x20      proto=0x11      chksum = 0x5885
        -- UDP --
                source port = 0x89      dest port = 0x89
                len = 0x4c      checksum = 0xa6a0
        -- DATA --
                00000014:                                       00 01 00 00  |
         ....
                00000024: 00 00 00 01 20 45 49 45 50 45 47 45 47 45 46 46  | ..
.. EIEPEGEGEFF
                00000034: 43 43 4e 46 41 45 44 43 41 43 41 43 41 43 41 43  | CC
NFAEDCACACACAC
                00000044: 41 43 41 41 41 00 00 20 00 01 c0 0c 00 20 00 01  | AC
AAA.. ..... ..
                00000054: 00 04 93 e0 00 06 60 00 01 02 03 04 00          | ..
....`......
--------- END OF PACKET ---------
```

This display lists the information as it appears in a packet.

The following is sample output from the **show debug** command:

```
show debug
debug icmp trace off
debug packet off
debug sqlnet off
```

Related Commands

mgcp	Configures additional support for the Media Gateway Control Protocol fixup (packet application inspection) and is used with the **fixup protocol mgcp** command.
show conn	Displays all active connections. There is an MGCP **show conn** option and connection flag, "g".
timeout	Sets the maximum idle time duration. (There is an MGCP timeout option.)

dhcpd

Configures the DHCP server.

[**no**] **dhcpd address** *ip1*[**-***ip2*] *if_name*

[**no**] **dhcpd auto_config** [**outside**]

[**no**] **dhcpd dns** *dns1* [*dns2*]

[**no**] **dhcpd wins** *wins1* [*wins2*]

[**no**] **dhcpd lease** *lease_length*

[**no**] **dhcpd domain** *domain_name*

[**no**] **dhcpd enable** *if_name*

[**no**] **dhcpd option 66 ascii** {*server_name* | *server_ip_str*}

[**no**] **dhcpd option 150 ip** *server_ip1* [*server_ip2*]

no dhcpd option *code*

[**no**] **dhcpd ping_timeout** *timeout*

[**no**] **debug dhcpd event**

[**no**] **debug dhcpd packet**

clear dhcpd [**binding|statistics**]

show dhcpd [**binding|statistics**]

Syntax Description

address *ip1* [*ip2*]	The IP pool address range. The size of the pool is limited to 32 addresses with a 10-user license and 128 addresses with a 50-user license on the PIX 501. The unlimited user license on the PIX 501 and all other PIX Firewall platforms support 256 addresses. If the address pool range is larger than 253 addresses, the netmask of the PIX Firewall interface cannot be a Class C address (for example, 255.255.255.0) and hence needs to be something larger, for example, 255.255.254.0.
auto_config	Enable PIX Firewall to automatically configure DNS, WINS and domain name values from the DHCP client to the DHCP server. If the user also specifies **dns**, **wins**, and **domain** parameters, then the CLI parameters overwrite the auto_config parameters.
binding	The binding information for a given server IP address and its associated client hardware address and lease length.
code	Specifies the DHCP option code, either 66 or 150.
dns *dns1* [*dns2*]	The IP addresses of the DNS servers for the DHCP client. Specifies that DNS A (address) resource records that match the static translation are rewritten. A second server address is optional.
domain *domain_name*	The DNS domain name. For example, **example.com**.
if_name	Specifies the interface on which to enable the DHCP server.
lease *lease_length*	The length of the lease, in seconds, granted to DHCP client from the DHCP server. The lease indicates how long the client can use the assigned IP address. The default is 3600 seconds. The minimum lease length is 300 seconds, and the maximum lease length is 2,147,483,647 seconds.
option 150	Specifies the TFTP server IP address(es) designated for Cisco IP Phones in dotted-decimal format. DHCP option 150 is site-specific; it gives the IP addresses of a list of TFTP servers.
option 66	Specifies the TFTP server IP address designated for Cisco IP Phones and gives the IP address or the host name of a single TFTP server.
outside	The **outside** interface of the firewall.
ping_timeout	Allows the configuration of the timeout value of a ping, in milliseconds, before assigning an IP address to a DHCP client.
server_ip(1,2)	Specifies the IP address(es) of a TFTP server.
server_ip_str	Specifies the TFTP server in dotted-decimal format, such as 1.1.1.1, but is treated as a character string by the PIX Firewall DHCP server.
server_name	Specifies an ASCII character string representing the TFTP server.
statistics	Statistical information, such as address pool, number of bindings, malformed messages, sent messages, and received messages.
wins *wins1* [*wins2*]	The IP addresses of the Microsoft NetBIOS name servers (WINS server). The second server address is optional.

Command Modes

Configuration mode.

Usage Guidelines

A DHCP server provides network configuration parameters to a DHCP client. Support for the DHCP server within the PIX Firewall means the PIX Firewall can use DHCP to configure connected clients. This DHCP feature is designed for the remote home or branch office that will establish a connection to an enterprise or corporate network. See the *Cisco PIX Firewall and VPN Configuration Guide* for information on how to implement the DHCP server feature into the PIX Firewall.

You must specify an interface name, *if_name*, for all DHCP server commands when using PIX Firewall software Version 6.3. In earlier software versions, only the inside interface could be configured as the DHCP server so there was no need to specify *if_name*.

Note The PIX Firewall DHCP server does not support **BOOTP** requests and **failover** configurations.

The **dhcpd address** *ip1*[**-***ip2*] *if_name* command specifies the DHCP server address pool. The address pool of a PIX Firewall DHCP server must be within the same subnet of the PIX Firewall interface that is enabled and you must specify the associated PIX Firewall interface with the *if_name*. In other words, the client must be physically connected to the subnet of a PIX Firewall interface. The size of the pool is limited to 32 addresses with a 10-user license and 128 addresses with a 50-user license on the PIX 501. The unlimited user license on the PIX 501 and all other PIX Firewall platforms support 256 addresses.

Note When the PIX Firewall responds to a DHCP client request, it uses the IP address of the interface where the request was received as the default gateway in the response. It uses the subnet mask on that interface for the subnet mask in its response.

Use caution with names that contain a "-" (dash) character because the **dhcpd address** command interprets the last (or only) "-" character in the name as a range specifier instead of as part of the name. For example, the **dhcpd address** command treats the name "`host-net2`" as a range from "`host`" to "`net2`". If the name is "`host-net2-section3`" then it is interpreted as a range from "`host-net2`" to "`section3`".

The **no dhcpd address** command removes the DHCP server address pool you configured.

The **dhcpd dns** command specifies the IP address(es) of the DNS server(s) for DHCP client. You have the option to specify two DNS servers. The **no dhcpd dns** command removes the DNS IP address(es) from your configuration.

The **dhcpd wins** command specifies the addresses of the WINS server for the DHCP client. The **no dhcpd dns** command removes the WINS server IP address(es) from your configuration.

The **dhcpd lease** command specifies the length of the lease in seconds granted to the DHCP client. This lease indicates how long the DHCP client can use the assigned IP address the DHCP granted. The **no dhcpd lease** command removes the lease length that you specified from your configuration and replaces this value with the default value of 3600 seconds.

The **dhcpd domain** command specifies the DNS domain name for the DHCP client. For example, **example.com**. The **no dhcpd domain** command removes the DNS domain server from your configuration.

The **dhcpd enable** *if_name* command enables the DHCP daemon to begin to listen for the DHCP client requests on the DHCP-enabled interface. The **no dhcpd enable** command disables the DHCP server feature on the specified interface.

DHCP must be enabled to use this command. Use the **dhcpd enable** *if_name* command to turn on DHCP.

Note The PIX Firewall DHCP server daemon does not support clients that are not directly connected to a firewall interface, and the interface must be configured to retrieve DHCP client information (with the **dhcprelay enable** *client_ifc* command).

The **dhcpd option 66 | 150** command retrieves TFTP server address information for Cisco IP Phone connections.

When a **dhcpd option** command request arrives at the PIX Firewall DHCP server, the PIX Firewall places the value(s) specified by the **dhcpd option 66 | 150** in the response.

Use the **dhcpd option** *code* command as follows:

- If the TFTP server for Cisco IP Phone connections is located on the inside interface, use the local IP address of the TFTP server in the **dhcpd option** command.
- If the TFTP server is located on a less secure interface, create a group of NAT, **global** and **access-list** command statements for the inside IP phones, and use the actual IP address of the TFTP server in the **dhcpd option** command.
- If the TFTP server is located on a more secure interface, create a group of **static** and **access-list** command statements for the TFTP server and use the global IP address of the TFTP server in the **dhcpd option** command.

The **show dhcpd** command displays **dhcpd** commands, binding and statistics information associated with all of the **dhcpd** commands.

The **clear dhcpd** command clears all of the **dhcpd** commands, binding, and statistics information.

The **debug dhcpd event** command displays event information about the DHCP server. The **debug dhcpd packet** command displays packet information about the DHCP server. Use the **no** form of the **debug dhcpd** commands to disable debugging.

Examples

The following partial configuration example shows how to use the **dhcpd address**, **dhcpd dns**, and **dhcpd enable** *if_name* commands to configure an address pool for the DHCP clients and a DNS server address for the DHCP client, and how to enable the **dmz** interface of the PIX Firewall for the DHCP server function.

```
dhcpd address 10.0.1.100-10.0.1.108 dmz
dhcpd dns 209.165.200.226
dhcpd enable dmz
```

The following partial configuration example shows how to define a DHCP pool of 253 addresses and use the **auto_config** command to configure the DNS, WINS, and DOMAIN parameters. Note that the **dmz** interface of the firewall is configured as the DHCP server, and the netmask of the **dmz** interface is 255.255.254.0:

```
ip address dmz 10.0.1.1 255.255.254.0
dhcpd address 10.0.1.2-10.0.1.254 dmz
dhcpd auto_config outside
dhcpd enable dmz
```

The following partial configuration example shows how to use three new features that are associated with each other: DHCP server, DHCP client, and PAT using interface IP to configure a PIX Firewall in a small office, home office (SOHO) environment with the **inside** interface as the DHCP server:

```
! use dhcp to configure the outside interface and default route
ip address outside dhcp setroute
! enable dhcp server daemon on the inside interface
ip address inside 10.0.1.2 255.255.255.0
dhcpd address 10.0.1.100-10.0.1.108 inside
dhcpd dns 209.165.201.2 209.165.202.129
dhcpd wins 209.165.201.5
dhcpd lease 3600
dhcpd domain example.com
dhcpd enable inside
! use outside interface IP as PAT global address
nat (inside) 1 0 0
global (outside) 1 interface
```

The following is sample output from the **show dhcpd** command:

```
pixfirewall(config)# show dhcpd
dhcpd address 10.0.1.100-10.0.1.108 inside
dhcpd lease 3600
dhcpd ping_timeout 750
dhcpd dns 209.165.201.2 209.165.202.129
dhcpd enable inside
```

The following is sample output from the **show dhcpd binding** command:

```
pixfirewall(config)# show dhcpd binding
IP Address Hardware Address Lease Expiration Type
10.0.1.100 0100.a0c9.868e.43 84985 seconds automatic
```

The following is sample output from the **show dhcpd statistics** command:

```
show dhcpd statistics
Address Pools 1
Automatic Bindings 1
Expired Bindings 1
Malformed messages 0

Message Received
BOOTREQUEST 0
DHCPDISCOVER 1
DHCPREQUEST 2
DHCPDECLINE 0
DHCPRELEASE 0
DHCPINFORM 0

Message Sent
BOOTREPLY 0
DHCPOFFER 1
DHCPACK 1
DHCPNAK 1
```

Related Commands

ip address	Configures the IP address and mask for an interface, or defines a local address pool.

dhcprelay

Configures the DHCP relay agent, which relays requests between the firewall interface of the DCHP server and DHCP clients on a different firewall interface.

[**no**] **dhcprelay enable** *client_ifc*

[**no**] **dhcprelay server** *dhcp_server_ip server_ifc*

[**no**] **dhcprelay setroute** *client_ifc*

[**no**] **dhcprelay timeout** *seconds*

[**clear|show**] **dhcprelay** [**statistics**]

Syntax Description

client_ifc	The name of the interface on which the DHCP relay agent accepts client requests.
dhcp_server_ip	The IP address of the DHCP server to which the DHCP relay agent forwards client requests.
enable	Enables the DHCP relay agent to accept DHCP requests from clients on the specified interface.
seconds	The number of seconds allowed for DHCP relay address negotiation.
server_ifc	The name of the firewall interface on which the DHCP server resides.
statistics	The DHCP relay statistics, incremented until a **clear dhcprelay statistics** command is issued.

Defaults

By default, the DHCP relay agent is disabled.

The default DHCP relay timeout value is 60 seconds.

Command Modes

Configuration mode. The **show dhcprelay** commands are also available in privileged mode.

Usage Guidelines

Use the **dhcprelay enable**, **dhcprelay server**, and **dhcprelay timeout** commands to configure the DHCP relay agent to relay requests between the firewall interface of the DCHP server and DHCP clients on a different firewall interface.

Note Use network extension mode for DHCP clients whose DHCP server is on the other side of an Easy VPN tunnel. Otherwise, if the DHCP client is behind a PIX Firewall VPN Easy Remote device connected to an Easy VPN Server using client mode, then the DHCP client will not be able to get a DHCP IP address from the DHCP server on the other side of the Easy VPN Server.

dhcprelay enable

For the firewall to start the DHCP relay agent with the **dhcprelay enable** *client_ifc* command, you must have a **dhcprelay server** command already in your configuration. Otherwise, the firewall displays an error message similar to the following:

```
DHCPRA:Warning - There are no DHCP servers configured!
       No relaying can be done without a server!
       Use the 'dhcprelay server <server_ip> <server_ifc>' command
```

The **dhcprelay enable** *client_ifc* command starts a DHCP server task on the specified interface. If this **dhcprelay enable** command is the first **dhcprelay enable** command to be issued, and there are **dhcprelay server** commands in the configuration, then the ports for the DHCP servers referenced are opened and the DHCP relay task starts.

When a **dhcprelay enable** *client_ifc* command is removed with a **no dhcprelay enable** *client_ifc* command, the DHCP server task for that interface stops. When the **dhcprelay enable** command being removed is the last **dhcprelay enable** command in the configuration, all of the ports for the servers specified in the **dhcprelay server** commands are closed and the DHCP relay task stops.

dhcprelay server

Add at least one **dhcprelay server** command to your firewall configuration before you enter a **dhcprelay enable** command or the firewall will issue an error message.

The **dhcprelay server** command opens a UDP port 67 on the specified interface for the specified server and starts the DHCP relay task as soon as a **dhcprelay enable** command is added to the configuration. If there is **no dhcprelay enable** command in the configuration, then the sockets are not opened and the DHCP relay task does not start.

When a **dhcprelay server** *dhcp_server_ip* [*server_ifc*] command is removed, the port for that server is closed. If the **dhcprelay server** command being removed is the last **dhcprelay server** command in the configuration, then the DHCP relay task stops.

dhcprelay setroute

The **dhcprelay setroute** *client_ifc* command enables you to configure the DHCP Relay Agent to change the first default router address (in the packet sent from the DHCP server) to the address of *client_ifc*. That is, the DHCP Relay Agent substitutes the address of the default router with the address of *client_ifc*.

If there is no default router option in the packet, the firewall adds one containing the address of *client_ifc*. This allows the client to set its default route to point to the firewall.

When the **dhcprelay setroute** *client_ifc* command is not configured (and there is a default router option in the packet) it passes through the firewall with the router address unaltered.

dhcprelay timeout

The **dhcprelay timeout** command sets the amount of time, in seconds, allowed for responses from the DHCP server to pass to the DHCP client through the relay binding structure.

no dhcprelay commands

The **no dhcprelay enable** *client_ifc* command removes the DHCP relay agent configuration for the interface specified by *client_ifc* only.

The **no dhcprelay server** *dhcp_server_ip* [*server_ifc*] command removes the DHCP relay agent configuration for the DHCP server and specified by *dhcp_server_ip* [*server_ifc*] only.

show dhcprelay

The **show dhcprelay** command displays the DHCP relay agent configuration, and the **show dhcprelay statistics** command displays counters for the packets relayed by the DHCP relay agent.

The **clear dhcprelay** command clears all DHCP relay configurations. The **clear dhcprelay statistics** command clears the **show dhcprelay statistics** counters.

Examples

The following example configures the DHCP relay agent for a DHCP server with the IP address of 10.1.1.1 on the outside interface of the firewall and client requests on the inside interface of the firewall, and sets the timeout value to 90 seconds:

```
pixfirewall(config)# dhcprelay server 10.1.1.1 outside
pixfirewall(config)# show dhcprelay
dhcprelay server 10.1.1.1 outside
dhcprelay timeout 50

pixfirewall(config)# dhcprelay timeout 60
pixfirewall(config)# show dhcprelay
dhcprelay server 10.1.1.1 outside
dhcprelay timeout 60

pixfirewall(config)# dhcprelay enable inside
pixfirewall(config)# show dhcprelay
dhcprelay server 10.1.1.1 outside
dhcprelay enable inside
dhcprelay timeout 60
```

The following example shows how to disable the DHCP relay agent if there is only one **dhcprelay enable** command in the configuration:

```
pixfirewall(config)# no dhcprelay enable
pixfirewall(config)# show dhcprelay
dhcprelay server 10.1.1.1 outside
dhcprelay timeout 60
```

The following is sample output from the **show dhcprelay statistics** command:

```
pixfirewall(config)# show dhcprelay statistics
Packets Relayed
BOOTREQUEST          0
DHCPDISCOVER         7
DHCPREQUEST          3
DHCPDECLINE          0
DHCPRELEASE          0
DHCPINFORM           0

BOOTREPLY            0
DHCPOFFER            7
DHCPACK              3
DHCPNAK              0
```

Related Commands

dhcpd	Controls the DHCP server feature.

disable

Exit privileged mode and return to unprivileged mode.

enable

disable

Syntax Description

enable	Enter this at the PIX Firewall command-line interface prompt to enter privileged mode.
disable	Enter this at the PIX Firewall command-line interface prompt to exit privileged mode.

Command Modes

Privileged mode.

Usage Guidelines

Use the **enable** command to enter privileged mode. The **disable** command exits privileged mode and returns you to unprivileged mode.

Examples

The following example shows how to enter privileged mode:

```
pixfirewall> enable
pixfirewall#
```

The following example shows how to exit privileged mode:

```
pixfirewall# disable
pixfirewall>
```

domain-name

Change the IPSec domain name.

domain-name *name*

Syntax Description

name	A domain name, up to 63 characters.

Command Modes

Configuration mode.

Usage Guidelines

The **domain-name** command lets you change the IPSec domain name.

Note The change of the domain name causes the change of the fully qualified domain name. Once the fully qualified domain name is changed, delete the RSA key pairs using the **ca zeroize rsa** command, and delete related certificates using the **no ca identity** *ca_nickname* command.

Examples The following example shows use of the **domain-name** command:

domain-name example.com

dynamic-map

View or delete a dynamic crypto map entry. To configure crypto dynamic map entries, see the **crypto dynamic-map** command.

clear dynamic-map

show dynamic-map

Syntax Description

dynamic-map	A dynamic crypto map entry.

Command Modes Configuration mode.

Usage Guidelines The **clear dynamic-map** command removes **dynamic-map** commands from the configuration. The **show dynamic-map** command lists the **dynamic-map** commands in the configuration.

Note The **dynamic-map** command is the same as the **crypto dynamic-map** command. Refer to the **crypto dynamic-map** command page for more information such as examples and other command options.

eeprom

This command applies only to PIX 525 models with serial numbers 44480380055 through 44480480044. Displays and updates the contents of the EEPROM non-volatile storage devices used for low-level Ethernet interface configuration information.

eeprom update

show eeprom

Syntax Description

eeprom update	Modifies the EEPROM register settings, if necessary, after checking the contents of EEPROM registers 6 and 10 to ensure they contain the hexadecimal values 0x4701 and 0x40c0, respectively. If these registers contain different values, then all EEPROM register settings, except the MAC address registers, which were not affected by the problem, are reset to the correct values.

Command Modes

Configuration mode.

Usage Guidelines

The **eeprom** commands added in Version 5.2(4) and higher fix a caveat (CSCds76768) involving corruption of the eeprom on the onboard Ethernet interfaces. For additional information, see the December 20, 2000 Field Notice, “Cisco Secure PIX Firewall: PIX-525 Ethernet EEPROM Programming Issue.” This field notice is available at the following website:

http://www.cisco.com/warp/public/770/fn13021.shtml

The problem is summarized as follows:

If you configure the onboard Ethernet interfaces (ethernet0 and ethernet1) on a PIX 525 with a serial number of 44480380055 through 44480480044 to full duplex, interface errors and throughput reductions may occur. If you configure the interfaces to half duplex or to auto-sense, the speed and duplex function normally without error.

The **eeprom** command is designed to fix the problem and performs the same function as the "eedisk" utility without requiring access to the ROM monitor mode. The two variants of the **eeprom** command are the **show eeprom** command and **eeprom update** command.

The **eeprom update** command performs the same function as the "eedisk" utility without requiring access to the ROM monitor mode, whereas the **show eeprom** command indicates whether the Ethernet EEPROM programming is correct or not.

The **show eeprom** command displays the current EEPROM setting, and the **eeprom update** command modifies the settings if necessary. If the **eeprom** command does update the EEPROM settings, a reboot of the PIX Firewall is recommended.

The **eeprom** command verifies the EEPROM register settings and updates them if they are not set to the recommended values. The **eeprom** command does not update the settings if they are correct and does not recommend a reboot unless the settings are changed.

The **eeprom update** command checks the contents of EEPROM registers 6 and 10 to ensure they contain the hexadecimal values 0x4701 and 0x40c0, respectively. If these registers contain different values, then all EEPROM register settings except the MAC address registers, which were not affected by the problem causing CSCds76768, are reset to the correct values.

Each register is 16 bits. The correct register values are as follows:

Register	Name	Value
Register 0 to 2	MAC address	Differs on each system (unique)
Register 3	Compatibility Bits	0x3
Register 5	Controller and connector type	0x201
Register 6	Onboard PHY type	0x4701

Register	Name	Value
Register 10	Onboard Prom ID	0x40C0
Register 12	Vendor ID, where 8086 is Intel	0x8086

Examples

The **show eeprom** command will display the current EEPROM register settings:

```
pix525# show eeprom
 eeprom settings for ifc0:
  reg0: 0x5000
  reg1: 0xfe54
  reg2: 0x65f6
  reg3: 0x3
  reg5: 0x201
  reg6: 0x4702
  reg10: 0x40c0
  reg12: 0x8086
 eeprom settings for ifc1:
  reg0: 0x5000
  reg1: 0xfe54
  reg2: 0x66f6
  reg3: 0x3
  reg5: 0x201
  reg6: 0x4702
  reg10: 0x40c0
  reg12: 0x8086reg12: 0x8086
```

If the command is run on a unit that is not a PIX 525, the following will be seen:

```
pix515# show eeprom
This unit is not a PIX-525.
Type help or '?' for a list of available commands.
```

If the update needs to be run on the PIX 525, the **eeprom update** command returns the following:

```
pix525# eeprom update
 eeprom settings on ifc0 are being reset to defaults:
  reg0: 0x5000
  reg1: 0xfe54
  reg2: 0x65f6
  reg3: 0x3
  reg5: 0x201
  reg6: 0x4701
  reg10: 0x40c0
  reg12: 0x8086
eeprom settings on ifc1 are being reset to defaults:
  reg0: 0x5000
  reg1: 0xfe54
  reg2: 0x66f6
  reg3: 0x3
  reg5: 0x201
  reg6: 0x4701
  reg10: 0x40c0
  reg12: 0x8086
*** WARNING! *** WARNING! *** WARNING! *** WARNING! ***
 The system should be restarted as soon as possible.
*** WARNING! *** WARNING! *** WARNING! *** WARNING! ***
```

If the update has been run successfully, the **eeprom** command output will appear as follows:

```
pix525# eeprom update
 eeprom settings on ifc0 are already up to date:
  reg0: 0x5000
  reg1: 0xfe54
  reg2: 0x65f6
  reg3: 0x3
  reg5: 0x201
  reg6: 0x4701
  reg10: 0x40c0
  reg12: 0x808
eeprom settings on ifc1 are already up to date:
  reg0: 0x5000
  reg1: 0xfe54
  reg2: 0x66f6
  reg3: 0x3
  reg5: 0x201
  reg6: 0x4701
  reg10: 0x40c0
  reg12: 0x80866
```

enable

Start privileged mode or access privilege levels.

enable [*priv_1evel*]

disable [*priv_1evel*]

enable password [*pw*] [**level** *priv_1evel*] [**encrypted**]

no enable password [**level** *priv_1evel*]

show enable

Syntax Description

enable	Specifies to activate a process, mode, or privilege level.
enable *priv_level*	Specifies to enable the privilege level, from 0 to 15.
encrypted	Specifies that the provided password is already encrypted.
level *priv_level*	Specifies to set the privilege level, from 0 to 15.
password	Specifies to configure privilege levels.
pw	The privilege level password string.

Command Modes

Unprivileged mode for **enable**, and configuration mode for **enable password**.

Usage Guidelines

The **enable** command starts privileged mode(s). The PIX Firewall prompts you for your privileged mode password. By default, a password is not required—press the **Enter** key at the Password prompt to start privileged mode. Use the **disable** command to exit privileged mode. Use the **enable password** command to change the password.

The **enable password** command changes the privileged mode password, for which you are prompted after you enter the **enable** command. When the PIX Firewall starts and you enter privileged mode, the password prompt appears. There is not a default password (press the **Enter** key at the Password prompt).

You can return the enable password to its original value (press the **Enter** key at prompt) by entering the following command:

```
pixfirewall# enable password
pixfirewall#
```

Note If you change the password, write it down and store it in a manner consistent with your site's security policy. Once you change this password, you cannot view it again. Also, ensure that all who access the PIX Firewall console are given this password.

Use the **passwd** command to set the password for Telnet access to the PIX Firewall console. The default **passwd** value is **cisco**.

See the **passwd** command page for more information.

If no privilege level name is specified, then the highest privilege level is assumed.

The **show enable** command displays the password configuration for privilege levels.

Examples

The following example shows how to start privileged mode with the **enable** command and then configuration mode with the **configure terminal** command.

```
pixfirewall> enable
Password:
pixfirewall# configure terminal
pixfirewall(config)#
```

The following examples show how to start privileged mode with the **enable** command, change the enable password with the **enable password** command, enter configuration mode with the **configure terminal** command, and display the contents of the current configuration with the **write terminal** command:

```
pixfirewall> enable
Password:
pixfirewall# enable password w0ttal1fe
pixfirewall# configure terminal
pixfirewall(config)# write terminal
Building configuration...
...
enable password 2oifudsaoid.9ff encrypted
...
```

The following example shows the use of the **encrypted** option:

```
enable password 1234567890123456 encrypted
show enable password
enable password 1234567890123456 encrypted

enable password 1234567890123456
show enable password
enable password feCkwUGktTCAgIbD encrypted
```

The following example shows how to configure enable passwords for levels other than the default level of 15:

```
pixfirewall(config)# enable password cisco level 10

pixfirewall(config)# show enable
enable password wC38a.EQklqK3ZqY level 10 encrypted
enable password 8Ry2YjIyt7RRXU24 encrypted

pixfirewall(config)# enable password wC38a.EQklqK3ZqY level 12 encrypted

pixfirewall(config)# show enable
enable password wC38a.EQklqK3ZqY level 10 encrypted
enable password wC38a.EQklqK3ZqY level 12 encrypted
enable password 8Ry2YjIyt7RRXU24 encrypted

pixfirewall(config)# no enable password level 12

pixfirewall(config)# show enable
enable password wC38a.EQklqK3ZqY level 10 encrypted
enable password 8Ry2YjIyt7RRXU24 encrypted

pixfirewall(config)# no enable password level 10

pixfirewall(config)# show enable
enable password 8Ry2YjIyt7RRXU24 encrypted
```

However, notice that defining privilege levels 10 and 12 does not change or remove the level 15 password.

established

Permit return connections on ports other than those used for the originating connection based on an established connection.

[no] established <est_protocol> <dport> [sport] [permitto <protocol> <port>[-<port>]] [permitfrom <protocol> <port>[-<port>]]

clear established

show established

Syntax Description

dest_port	Specifies the destination port to use for the established connection lookup. This is the originating traffic's destination port and may be specified as 0 if the protocol does not specify which destination port(s) will be used. Use wildcard ports (0) only when necessary.
permitfrom	Used to specify the return traffic's protocol and from which source port(s) the traffic will be permitted.
permitto	Used to specify the return traffic's protocol and to which destination port(s) the traffic will be permitted.
src_port	Specifies the source port to use for the established connection lookup. This is the originating traffic's source port and may be specified as 0 if the protocol does not specify which source port(s) will be used. Use wildcard ports (0) only when necessary.

Command Modes Configuration mode.

Usage Guidelines The **established** command allows outbound connections return access through the PIX Firewall. This command works with two connections, an original connection outbound from a network protected by the PIX Firewall and a return connection inbound between the same two devices on an external host.

The first protocol, destination port, and optional source port specified are for the initial outbound connection. The **permitto** and **permitfrom** options refine the return inbound connection.

Note We recommend that you always specify the **established** command with the **permitto** and **permitfrom** options. Without these options, the use of the **established** command opens a security hole that can be exploited for attack of your internal systems. See the "Security Problem" section that follows for more information.

The **permitto** option lets you specify a new protocol or port for the return connection at the PIX Firewall.

The **permitfrom** option lets you specify a new protocol or port at the remote server.

The **no established** command disables the **established** feature.

The **clear established** command removes all **establish** command statements from your configuration.

Note For the **established** command to work properly, the client must listen on the port specified with the **permitto** option.

You can use the **established** command with the **nat 0** command statement (where there are no **global** command statements).

Note The **established** command cannot be used with Port Address Translation (PAT).

The **established** command works as shown in the following format:

```
established A B C permitto D E permitfrom D F
```

This command works as though it were written "If there exists a connection between two hosts using protocol A from src port B destined for port C, permit return connections through the PIX Firewall via protocol D (D can be different from A), if the source port(s) correspond to F and the destination port(s) correspond to E."

For example:

```
established tcp 6060 0 permitto tcp 6061 permitfrom tcp 6059
```

In this case, if a connection is started by an internal host to an external host using TCP source port 6060 and any destination port, the PIX Firewall permits return traffic between the hosts via TCP destination port 6061 and TCP source port 6059.

For example:

```
established udp 0 6060 permitto tcp 6061 permitfrom tcp 1024-65535
```

In this case, if a connection is started by an internal host to an external host using UDP destination port 6060 and any source port, the PIX Firewall permits return traffic between the hosts via TCP destination port 6061 and TCP source port 1024-65535.

Security Problem

The **established** command has been enhanced to optionally specify the destination port used for connection lookups. Only the source port could be specified previously with the destination port being 0 (a wildcard). This addition allows more control over the command and provides support for protocols where the destination port is known, but the source port is not.

The **established** command can potentially open a large security hole in the PIX Firewall if not used with discretion. Whenever you use this command, if possible, also use the **permitto** and **permitfrom** options to indicate ports to which and from which access is permitted. Without these options, external systems to which connections are made could make unrestricted connections to the internal host involved in the connection. The following are examples of potentially serious security violations that could be allowed when using the **established** command.

For example:

```
established tcp 0 4000
```

In this example, if an internal system makes a TCP connection to an external host on port 4000, then the external host could come back in on any port using any protocol:

`established tcp 0 0` **(Same as previous releases established tcp 0** command**.)**

Examples

The following example occurs when a local host 10.1.1.1 starts a TCP connection on port 9999 to a foreign host 209.165.201.1. The example allows packets from the foreign host 209.165.201.1 on port 4242 back to local host 10.1.1.1 on port 5454.

```
established tcp 9999 permitto tcp 5454 permitfrom tcp 4242
```

The next example allows packets from foreign host 209.165.201.1 on any port back to local host 10.1.1.1 on port 5454:

```
established tcp 9999 permitto tcp 5454
```

XDMCP Support

PIX Firewall now provides support for XDMCP (X Display Manager Control Protocol) with assistance from the **established** command.

XDMCP is on by default, but will not complete the session unless the **established** command is used.

For example:

```
established tcp 0 6000 permitto tcp 6000 permitfrom tcp 1024-65535
```

This enables the internal XDMCP equipped (UNIX or ReflectionX) hosts to access external XDMCP equipped XWindows servers. UDP/177 based XDMCP negotiates a TCP based XWindows session and subsequent TCP back connections will be permitted. Because the source port(s) of the return traffic is unknown, the *src_port* field should be specified as 0 (wildcard). The destination port, *dest_port*, will typically be 6000; the well-known XServer port. The *dest_port* should be 6000 + *n*; where *n* represents the local display number. Use the following UNIX command to change this value.

setenv DISPLAY *hostname:displaynumber.screennumber*

The **established** command is needed because many TCP connections are generated (based on user interaction) and the source port for these connection is unknown. Only the destination port will be static. The PIX Firewall does XDMCP fixups transparently. No configuration is required, but the **established** command is necessary to accommodate the TCP session. Be advised that using applications like this through the PIX Firewall may open up security holes. The XWindows system has been exploited in the past and newly introduced exploits are likely to be discovered.

exit

Exit an access mode.

exit

enable

Syntax Description

exit	Exits the current command mode.
enable	Enables privileged mode.

Command Modes

All modes.

Usage Guidelines

Use the **exit** command to exit from an access mode. This command is the same as the **quit** command.

Examples

The following example shows how to exit configuration mode and then privileged mode:

```
pixfirewall(config)# exit
pixfirewall# exit
pixfirewall>
```

failover

Enable or disable the PIX Firewall failover feature on a standby PIX Firewall.

[**no**] **failover** [**active**]

[**no**] **failover ip address** *if_name ip_address*

[**no**] **failover lan unit primary | secondary**

[**no**] **failover lan interface** *lan_if_name*

[**no**] **failover lan key** *key_secret*

[**no**] **failover lan enable**

[**no**] **failover link** [*stateful_if_name*]

[**no**] **failover mac address** *mif_name act_mac stn_mac*

[**no**] **failover poll** *seconds*

[**no**] **failover replicate http**

failover reset

show failover [**lan** [**detail**]]

Syntax Description

act_mac	The interface MAC address for the active PIX Firewall.
active	Make a PIX Firewall the active unit. Use this command when you need to force control of the connection back to the unit you are accessing, such as when you want to switch control back from a unit after you have fixed a problem and want to restore service to the primary unit. Either enter the **no failover active** command on the secondary unit to switch service to the primary or the **failover active** command on the primary unit.
detail	Displays LAN-based failover configuration information.
enable	Enables LAN-based failover; otherwise, serial cable failover is used.
if_name	The interface name for the failover IP address.
ip_address	The IP address used by the standby unit to communicate with the active unit. Use this IP address with the **ping** command to check the status of the standby unit. This address must be on the same network as the system IP address. For example, if the system IP address is 192.159.1.3, set the failover IP address to 192.159.1.4.
key	Enables encryption and authentication of LAN-based failover messages between PIX Firewalls.
key_secret	The shared secret key.
lan	Specifies LAN-based failover.
lan interface *lan_if_name*	The name of the firewall interface dedicated to LAN-based failover. The interface name of a VLAN logical interface cannot be used for *lan_if_name*.
link	Specify the interface where a Fast Ethernet or Gigabit LAN link is available for Stateful Failover. A VLAN logical interface cannot be used.
mif_name	The name of the interface to set the MAC address.
poll *seconds*	Specify how long failover waits before sending special failover "hello" packets between the primary and standby units over all network interfaces and the failover cable. The default is 15 seconds. The minimum value is 3 seconds and the maximum is 15 seconds. Set to a lower value for Stateful Failover. With a faster poll time, PIX Firewall can detect failure and trigger failover faster. However, faster detection may cause unnecessary switchovers when the network is temporarily congested or a network card starts slowly.
primary	Specifies the primary PIX Firewall to use for LAN-based failover.
replicate http	The [**no**] **failover replicate http** command allows the stateful replication of HTTP sessions in a Stateful Failover environment. The **no** form of this command disables HTTP replication in a Stateful Failover configuration. When HTTP replication is enabled, the **show failover** command displays the **failover replicate http** command configuration.
reset	Force both units back to an unfailed state. Use this command once the fault has been corrected. The **failover reset** command can be entered from either unit, but it is best to always enter commands at the active unit. Entering the **failover reset** command at the active unit will "unfail" the standby unit.
secondary	Specifies the secondary PIX Firewall to use for LAN-based failover.

stateful_if_name	In addition to the failover cable, a dedicated Fast Ethernet or Gigabit LAN link is required to support Stateful Failover. The interface name of a VLAN logical interface cannot be used for *stateful_if_name*.
stn_mac	The interface MAC address for the standby PIX Firewall.

Command Modes

Configuration mode.

Usage Guidelines

The default failover setup uses serial cable failover. LAN-based failover requires explicit LAN-based failover configuration. Additionally, for LAN-based failover, you must install a dedicated 100 Mbps or Gigabit Ethernet, full-duplex VLAN switch connection for failover operations. Failover is not supported using a crossover Ethernet cable between two PIX Firewall units.

Note The PIX 506/506E cannot be used for failover in any configuration.

The primary unit in the PIX 515/515E, PIX 525, or PIX 535 failover pair must have an Unrestricted (UR) license. The secondary unit can have Failover (FO) or UR license. However, the failover pair must be two otherwise identical units with the same PIX Firewall hardware and software.

For a Stateful Failover link, use the **mtu** command to set the interface maximum transmission unit (MTU) to 1500 bytes or greater.

For serial cable failover, use the **failover** command without an argument after you connect the optional failover cable between your primary PIX Firewall and a secondary PIX Firewall. The default configuration has failover enabled. Enter **no failover** in the configuration file for the PIX Firewall if you will not be using the failover feature. Use the **show failover** command to verify the status of the connection and to determine which unit is active.

For LAN-based failover, use the **failover lan** commands. The **show failover lan** command displays LAN-based failover information (only), and **show failover lan detail** supplies debugging information for your LAN-based failover configuration.

Note Refer to the *Cisco PIX Firewall and VPN Configuration Guide* for configuration information.

For failover, the PIX Firewall requires that you configure any unused interfaces with one of the following methods:

- Shutdown the interface and do not configure its IP or failover IP address. If these addresses are configured, use the **no ip address** and **no failover ip address** commands to remove the configuration.
- Configure the interface like other interfaces but use a cross-over Ethernet cable to connect the interface to the Standby unit. Do not connect the interface to an external switch or hub device.

Set the speed of the Stateful Failover dedicated interface to 100full for a Fast Ethernet interface or 1000fullsx for a Gigabit Ethernet interface.

Use the **failover active** command to initiate a failover switch from the standby unit, or the **no failover active** command from the active unit to initiate a failover switch. You can use this feature to return a failed unit to service, or to force an active unit off line for maintenance. Because the standby unit does not keep state information on each connection, all active connections will be dropped and must be re-established by the clients.

Use the **failover link** command to enable Stateful Failover. Enter the **no failover link** command to disable the Stateful Failover feature.

If a failover IP address has not been entered, the **show failover** command will display 0.0.0.0 for the IP address, and monitoring of the interfaces will remain in "waiting" state. A failover IP address must be set for failover to work.

The **failover mac address** command enables you to configure a virtual MAC address for a PIX Firewall failover pair. The **failover mac address** command sets the PIX Firewall to use the virtual MAC address stored in the PIX Firewall configuration after failover, instead of obtaining a MAC address by contacting its failover peer. This enables the PIX Firewall failover pair to maintain the correct MAC addresses after failover. If a virtual MAC address is not specified, the PIX Firewall failover pair uses the burned in network interface card (NIC) address as the MAC address. However, the **failover mac address** command is unnecessary (and therefore cannot be used) on an interface configured for LAN-based failover because the **failover lan interface** *lan_if_name* command does not change the IP and MAC addresses when failover occurs.

When adding the **failover mac address** command to your configuration, it is best to configure the virtual MAC address, save the configuration to Flash memory, and then reload the PIX Firewall pair. If the virtual MAC address is added when there are active connections, then those connections will stop. Also, you must write the complete PIX Firewall configuration, including the **failover mac address** command, into the Flash memory of the secondary PIX Firewall for the virtual MAC addressing to take effect.

The **failover poll** *seconds* command lets you determine how long failover waits before sending special failover "hello" packets between the primary and standby units over all network interfaces and the failover cable. The default is 15 seconds. The minimum value is 3 seconds and the maximum is 15 seconds. Set to a lower value for Stateful Failover. With a faster poll time, PIX Firewall can detect failure and trigger failover faster. However, faster detection may cause unnecessary switchovers when the network is temporarily congested or a network card starts slowly.

When a failover cable connects two PIX Firewall units, the **no failover** command now disables failover until you enter the **failover** command to explicitly enable failover. Previously, when the failover cable connected two PIX Firewall units and you entered the **no failover** command, failover would automatically re-enable after 15 seconds.

You can also view the information from the **show failover** command using SNMP. Refer to the *Cisco PIX Firewall and VPN Configuration Guide* for more information on configuring failover.

Usage Notes

1. LAN-based failover requires a dedicated interface, but the same interface can also be used for Stateful Failover. However, the interface needs enough capacity to handle both the LAN-based failover and Stateful Failover traffic; otherwise, use two separate dedicated interfaces.
2. If you reboot the PIX Firewall without entering the **write memory** command and the failover cable is connected, failover mode automatically enables.

Examples

Serial Cable (Default) Failover

The following sample output shows that failover is enabled, and that the primary unit state is active:

```
show failover
pixfirewall (config)# show failover
       Failover On
       Cable status:Normal
       Reconnect timeout 0:00:00
       Poll frequency 15 seconds
       Last Failover at: 18:32:16 UTC Mon Apr 7 2003
       failover replication http
```

```
        This host:Secondary - Standby
                Active time:0 (sec)
                Interface FailLink (209.165.201.6):Normal
                Interface 4th (209.165.200.230):Normal
                Interface int5 (209.165.200.226):Normal
                Interface intf2 (192.168.1.1):Normal
                Interface outside (209.165.200.225):Normal
                Interface inside (10.1.1.4):Normal
        Other host:Primary - Active
                Active time:242145 (sec)
                Interface FailLink (172.16.31.1):Normal
```

The rest of command output is omitted.

The "Cable status" has these values:

- Normal—Indicates that the active unit is working and that the standby unit is ready.
- Waiting—Indicates that monitoring of the other unit's network interfaces has not yet started.
- Failed—Indicates that the PIX Firewall has failed.

The "Stateful Obj" has these values:

- Xmit—Indicates the number of packets transmitted.
- Xerr—Indicates the number of transmit errors.
- Rcv—Indicates the number of packets received.
- Rcv—Indicates the number of receive errors.

Each row is for a particular object static count:

- General—The sum of all stateful objects.
- Sys cmd—Refers to logical update system commands, such as **login** or **stay alive**.
- Up time—The value for PIX Firewall up time which the active PIX Firewall unit will pass on to the standby unit.
- Xlate—The PIX Firewall translation information.
- Tcp conn—The PIX Firewall dynamic TCP connection information.
- Udp conn—The PIX Firewall dynamic UDP connection information.
- ARP tbl—The PIX Firewall dynamic ARP table information.
- RIF tbl—The dynamic router table information.

The Standby Logical Update Statistics output displayed when you use the **show failover** command only describes Stateful Failover. The "xerrs" value does not indicate an error in failover, but rather the number of packet transmit errors.

You can view the IP addresses of the standby unit with the **show ip address** command:

```
show ip address
System IP Addresses:
        ip address outside 209.165.201.2 255.255.255.224
        ip address inside 192.168.2.1 255.255.255.0
        ip address perimeter 192.168.70.3 255.255.255.0
Current IP Addresses:
        ip address outside 209.165.201.2 255.255.255.224
        ip address inside 192.168.2.1 255.255.255.0
        ip address perimeter 192.168.70.3 255.255.255.0
```

The Current IP Addresses are the same as the System IP Addresses on the failover active unit. When the primary unit fails, the Current IP Addresses become those of the standby unit.

LAN-Based Failover

To make sure LAN-based failover starts properly, follow these configuration steps:

Step 1 Configure the primary PIX Firewall unit before connecting the failover LAN interface.

Step 2 Save the primary unit configuration to Flash memory.

Step 3 Configure the PIX Firewall secondary unit using the appropriate **failover lan** commands before connecting the LAN-based failover interface.

Step 4 Save the secondary unit configuration to Flash memory.

Step 5 Reboot both units and connect the LAN-based failover interfaces to the designated failover switch, hub, or VLAN.

Step 6 If any item in a **failover lan** command needs to be changed, then disconnect the LAN-based failover interface, and repeat the preceeding steps.

Note When properly configured, the LAN-based failover configurations for your primary and secondary PIX Firewall units should be different, reflecting which is primary and which is secondary.

The following example outlines how to configure LAN-based failover between two PIX Firewall units.

Primary PIX Firewall configuration:

```
    :
pix(config)# nameif ethernet0 outside security0
pix(config)# nameif ethernet1 inside security100
pix(config)# nameif ethernet2 stateful security20
pix(config)# nameif ethenret3 lanlink security30
    :
pix(config)#interface ethernet0 100full
pix(config)#interface ethernet1 100full
pix(config)#interface ethernet2 100full
pix(config)#interface ethenret3 100full
pix(config)#interface ethernet4 100full
    :
pix(config)# ip address outside 172.23.58.70 255.255.255.0
pix(config)# ip address inside   10.0.0.2 255.255.255.0
pix(config)# ip address stateful 10.0.1.2 255.255.255.0
pix(config)# ip address lanlink 10.0.2.2 255.255.255.0
pix(config)# failover ip address outside 172.23.58.51
pix(config)# failover ip address inside 10.0.0.4
pix(config)# failover ip address stateful 10.0.1.4
pix(config)# failover ip address lanlink  10.0.2.4
pix(config)# failover
pix(config)# failover poll 15
pix(config)# failover lan unit primary
pix(config)# failover lan interface lanlink
pix(config)# failover lan key 12345678
pix(config)# failover lan enable
    :
```

Secondary PIX Firewall configuration:

```
Pix2(config)# nameif  ethernet3 lanlink security30
pix2(config)# interface ethernet3 100full
```

```
pix2(config)# ip address lanlink 10.0.2.2 255.255.255.0
pix2(config)# failover ip address lanlink  10.0.2.4
pix2(config)# failover
pix2(config)# failover lan unit secondary           (optional)
pix2(config)# failover lan interface lanlink
pix2(config)# failover lan key 12345678
pix2(config)# failover lan enable
```

The following example illustrates how to use the **failover mac address** command:

```
ip address outside 172.23.58.50 255.255.255.224
ip address inside 192.168.2.11 255.255.255.0
ip address intf2 192.168.10.11 255.255.255.0
failover
failover ip address outside 172.23.58.51
failover ip address inside 192.168.2.12
failover ip address intf2 192.168.10.12
failover mac address outside 00a0.c989.e481 00a0.c969.c7f1
failover mac address inside 00a0.c976.cde5 00a0.c922.9176
failover mac address intf2 00a0.c969.87c8 00a0.c918.95d8
failover link intf2
            ...:
```

The output of the **show failover** command includes a section for LAN-based failover if it is enabled as follows:

```
pix(config)# show failover
Failover On
Cable status: Unknown
Reconnect timeout 0:00:00
Poll frequency 15 seconds
Last Failover at: 18:32:16 UTC Mon Apr 7 2003
            This host: Primary - Standby
                        Active time: 255 (sec)
                        Interface outside (192.168.1.232): Normal
                        Interface inside (192.168.5.2): Normal
            Other host: Secondary - Active
                        Active time: 256305 (sec)
                        Interface outside (192.168.1.231): Normal
                        Interface inside (192.168.5.1): Normal

Stateful Failover Logical Update Statistics
            Link : Unconfigured.

Lan Based Failover is Active
            interface dmz (209.165.200.226): Normal, peer (209.165.201.1): Normal
```

The **show failover lan** command displays only the LAN-based failover section, as follows:

```
pix(config)# show failover lan
Lan Based Failover is Active
            interface dmz (209.165.200.226): Normal, peer (209.165.201.1): Normal
```

The **show failover lan detail** command is used mainly for debugging purposes and displays information similar to the following:

```
pix(config)# show failover lan detail
Lan Failover is Active
This Pix is Primary
Command Interface is dmz
Peer Command Interface IP is 209.165.201.1
My interface status is 0x1
Peer interface status is 0x1
Peer interface downtime is 0x0
Total msg send: 103093, rcvd: 103031, droped: 0, retrans: 13, send_err: 0
Total/Cur/Max of 51486:0:5 msgs on retransQ
...
LAN FO cmd queue, count: 0, head: 0x0, tail: 0x0
Failover config state is 0x5c
Failover config poll cnt is 0
Failover pending tx msg cnt is 0
Failover Fmsg cnt is 0
   :
```

filter

Enable, disable, or view URL, FTP, HTTPS, Java, and ActiveX filtering

[**no**] **filter activex** *port local_ip mask foreign_ip mask*

[**no**] **filter ftp** *dest-port* | except *local_ip local_mask foreign_ip foreign_mask* [**allow**] [**interact-block**]

[**no**] **filter java** *port*[**-***port*]| except *local_ip mask foreign_ip mask*

[**no**] **filter https** *dest-port* | except *local_ip local_mask foreign_ip foreign_mask* [**allow**]

[**no**] **filter url** [**http** | *port*[**-***port*]] except *local_ip local_mask foreign_ip foreign_mask* [**allow**] [**proxy-block**] [**longurl-truncate | longurl-deny**] [**cgi-truncate**]

[**no**] **filter url except** *local_ip local_mask foreign_ip foreign_mask*

[**no**] **filter url** *port* | **except** *local_ip mask foreign_ip mask* [**allow**] [**proxy-block**] [**longurl-truncate | longurl-deny**] [**cgi-truncate**]

clear filter

show filter

Syntax Description

activex	Block inbound ActiveX, and other HTML <object> tags from outbound packets.
allow	**filter url** only: When the server is unavailable, let outbound connections pass through the firewall without filtering. If you omit this option, and if the N2H2 or Websense server goes off line, PIX Firewall stops outbound port 80 (Web) traffic until the N2H2 or Websense server is back on line.
cgi_truncate	Sends a CGI script as an URL.
dest-port	The destination port number.

except	Creates an exception to a previously specified set of IP addresses.
foreign_ip	The IP address of the lowest security level interface to which access is sought. You can use **0.0.0.0** (or in shortened form, **0**) to specify all hosts.
foreign_mask	Network mask of *foreign_ip*. Always specify a specific mask value. You can use **0.0.0.0** (or in shortened form, **0**) to specify all hosts.
ftp	Enables File Transfer Protocol (FTP) filtering. Available with Websense URL filtering only.
http	Specifies port 80. You can enter **http** or **www** instead of 80 to specify port 80.)
https	Enables HTTPS filtering. Available with Websense URL filtering only.
interact-block	Prevents users from connecting to the FTP server through an interactive FTP program.
java	Specifies to filter out Java applets returning from an outbound connection.
local_ip	The IP address of the highest security level interface from which access is sought. You can set this address to **0.0.0.0** (or in shortened form, **0**) to specify all hosts.
local_mask	Network mask of *local_ip*. You can use **0.0.0.0** (or in shortened form, **0**) to specify all hosts.
longurl-deny	Denies the URL request if the URL is over the URL buffer size limit or the URL buffer is not available.
longurl-truncate	Sends only the originating host name or IP address to the Websense server if the URL is over the URL buffer limit.
mask	Any mask.
port	The port that receives Internet traffic on the PIX Firewall. Typically, this is port 80, but other values are accepted. The **http** or **url** literal can be used for port 80.
proxy-block	Prevents users from connecting to an HTTP proxy server.
url	Filter Universal Resource Locators (URLs) from data moving through the PIX Firewall.

Command Modes

Configuration mode.

Usage Guidelines

The **clear filter** command removes all **filter** commands from the configuration.

filter activex

The **filter activex** command filters out ActiveX, Java applets, and other HTML <object> usages from outbound packets. ActiveX controls, formerly known as OLE or OCX controls, are components you can insert in a web page or other application. These controls include custom forms, calendars, or any of the extensive third-party forms for gathering or displaying information.

As a technology, it creates many potential problems for the network clients including causing workstations to fail, introducing network security problems, or be used to attack servers.

This feature blocks the HTML <object> tag and comments it out within the HTML web page.

Note The <object> tag is also used for Java applets, image files, and multimedia objects, which will also be blocked by the **filter activex** command. If the <object> or </object> HTML tags split across network packets or if the code in the tags is longer than the number of bytes in the MTU, the PIX Firewall cannot block the tag.

ActiveX blocking does not occur when users access an IP address referenced by the **alias** command.

To specify that all outbound connections have ActiveX blocking, use the following command:

```
filter activex 80 0 0 0 0
```

This command specifies that the ActiveX blocking applies to Web traffic on port 80 from any local host and for connections to any foreign host.

filter java

The **filter java** command filters out Java applets that return to the PIX Firewall from an outbound connection. The user still receives the HTML page, but the web page source for the applet is commented out so that the applet cannot execute. Use 0 for the *local_ip* or *foreign_ip* IP addresses to mean all hosts.

Note If Java applets are known to be in <object> tags, use the **filter activex** command to remove them.

To specify that all outbound connections have Java applet blocking, use the following command:

```
filter java 80 0 0 0 0
```

This command specifies that the Java applet blocking applies to Web traffic on port 80 from any local host and for connections to any foreign host.

filter url

The **filter url** command lets you prevent outbound users from accessing World Wide Web URLs that you designate using the N2H2 or Websense filtering application.

Note The **url-server** command must be configured before issuing the **filter** command for HTTPS and FTP, and if all URL servers are removed from the server list, then all **filter** commands related to URL filtering are also removed.

The **allow** option to the **filter** command determines how the PIX Firewall behaves in the event that the N2H2 or Websense server goes off line. If you use the **allow** option with the **filter** command and the N2H2 or Websense server goes offline, port 80 traffic passes through the PIX Firewall without filtering. Used without the **allow** option and with the server off line, PIX Firewall stops outbound port 80 (Web) traffic until the server is back on line, or if another URL server is available, passes control to the next URL server.

Note With the **allow** option set, PIX Firewall now passes control to an alternate server if the N2H2 or Websense server goes off line.

The N2H2 or Websense server works with the PIX Firewall to deny users from access to websites based on the company security policy.

Websense protocol Version 4 enables group and username authentication between a host and a PIX Firewall. The PIX Firewall performs a username lookup, and then Websense server handles URL filtering and username logging.

The N2H2 server must be a Windows workstation (2000, NT, or XP), running an IFP Server, with a recommended minimum of 512 MB of RAM. Also, the long URL support for the N2H2 service is capped at 3 KB, less than the cap for Websense.

Websense protocol Version 4 contains the following enhancements:

- URL filtering allows the PIX Firewall to check outgoing URL requests against the policy defined on the Websense server.
- Username logging tracks username, group, and domain name on the Websense server.
- Username lookup enables the PIX Firewall to use the user authentication table to map the host's IP address to the username.

Follow these steps to filter URLs:

Step 1 Designate an N2H2 or Websense server with the appropriate vendor-specific form of the **url-server** command.

Step 2 Enable filtering with the **filter** command.

Step 3 If needed, improve throughput with the **url-cache** command. However, this command does not update Websense logs, which may affect Websense accounting reports. Accumulate Websense run logs before using the **url-cache** command.

Step 4 Use the **show url-cache stats** and the **show perfmon** commands to view run information.

Information on Websense is available at the following website:

http://www.websense.com/

Examples

The following example filters all outbound HTTP connections except those from the 10.0.2.54 host:

```
url-server (perimeter) host 10.0.1.1
filter url 80 0 0 0 0
filter url except 10.0.2.54 255.255.255.255 0 0
```

The following example blocks all outbound HTTP connections destined to a proxy server that listens on port 8080:

```
filter url 8080 0 0 0 0 proxy-block
```

fixup protocol

Modifies PIX Firewall protocol fixups to add, delete, or change services and feature defaults.

fixup protocol ctiqbe 2748

[no] fixup protocol dns [maximum-length *length*]

fixup protocol esp-ike

fixup protocol ftp [**strict**] [*port*]

fixup protocol http [*port*[**-***port*]

fixup protocol h323 {**h225 | ras**} *port* [**-***port*]

fixup protocol icmp error

fixup protocol ils [*port*[**-***port*]]

[**no**] **fixup protocol mgcp** [*port*[**-***port*]]

fixup protocol pptp 1723

fixup protocol rsh [**514**]

fixup protocol rtsp [*port*]

fixup protocol sip [*port*[**-***port*]

[**no**] **fixup protocol sip udp 5060**

fixup protocol skinny [*port*[**-***port*]

fixup protocol smtp [*port*[**-***port*]]

fixup protocol snmp [161[-162]]

fixup protocol sqlnet [*port*[**-***port*]]

fixup protocol tftp [*port*[**-***port*]]

no fixup protocol [*protocol_name*] [*port*]

clear fixup

show ctiqbe

show fixup

show fixup protocol *protocol* [*protocol*]

show conn state [*protocol*]

show h225

show h245

show h323-ras

show mgcp

show sip

show skinny

show timeout *protocol*

Syntax Description

ctiqbe	Enables the Computer Telephony Interface Quick Buffer Encoding (CTIQBE) fixup. Used with Cisco TAPI/JTAPI applications.
dns	Enables the DNS fixup.
esp-ike	Enables PAT for Encapsulating Security Payload (ESP), single tunnel.
fixup protocol ils	Provides support for Microsoft NetMeeting, SiteServer, and Active Directory products that use LDAP to exchange directory information with an ILS server.
fixup protocol *protocol* [*protocol*] [*port*[**-***port*]]	Modifies PIX Firewall protocol fixups to add, delete, or change services and feature defaults.
ftp	Specifies to change the ftp port number.
h323 h225	Specifies to use H.225, the ITU standard that governs H.225.0 session establishment and packetization, with H.323. H.225.0 actually describes several different protocols: RAS, use of Q.931, and use of RTP.
h323 ras	Specifies to use RAS with H.323 to enable dissimilar communication devices to communicate with each other. H.323 defines a common set of CODECs, call setup and negotiating procedures, and basic data transport methods.
http [*port*[**-***port*]	The default port for HTTP is 80. Use the *port* option to change the HTTP port, or the *port-port* option to specify a range of HTTP ports.
ils	Specifies the Internet Locator Service. The default port is TCP LDAP server port 389.
dns **maximum-length** *length*	Specifies the maximum DNS packet length allowed. Default is 512 bytes.
mgcp	Enables the Media Gateway Control Protocol (MGCP) fixup. (Use the **mgcp** command to configure additional support for the MGCP fixup.)
no	Disables the fixup of a protocol by removing all fixups of the protocol from the configuration using the **no fixup** command. After removing all fixups for a protocol, the **no fixup** form of the command or the default port is stored in the configuration.
port	The port on which to enable the fixup (application inspection). You can use port numbers or supported port literals. The default ports are: TCP 21 for **ftp**, TCP LDAP server port 389 for **ils**, TCP 80 for **http**, TCP 1720 for **h323 h225**, UDP 1718-1719 for **h323 ras**, TCP 514 for **rsh**, TCP 554 for **rtsp**, TCP 2000 for **skinny**, TCP 25 for **smtp**, TCP 1521 for **sqlnet**, TCP 5060 for **sip**, and UDP 69 for TFTP. The default port value for **rsh** cannot be changed, but additional port statements can be added. See the "Ports" section in Chapter 2, "Using PIX Firewall Commands" for a list of valid port literal names. The port over which the designated protocol travels.
*port***-***port*	Specifies a port range.
pptp	Enables Point-to-Point Tunneling Protocol (PPTP) application inspection. The default port is 1723.
protocol	Specifies the protocol to fix up.
protocol_name	The protocol name.
ras	Registration, admission, and status (RAS) is a signaling protocol that performs registration, admissions, bandwidth changes, status, and disengage procedures between the VoIP gateway and the gatekeeper.

sip	Enable or change the port assignment for the Session Initiation Protocol (SIP) for Voice over IP TCP connections. UDP SIP is on by default and can be disabled and the port assignment is nonconfigurable. PIX Firewall Version 6.2 introduced PAT support for SIP.
skinny	Enable SCCP application inspection. The default port is **2000**. SCCP protocol supports IP telephony and can coexist in an H.323 environment. An application layer ensures that all SCCP signaling and media packets can traverse the PIX Firewall and interoperate with H.323 terminals. Skinny is the short name form for Skinny Client Control Protocol (SCCP).
snmp	Disabled by default. Enables SNMP inspection if enabled.
strict	Prevent web browsers from sending embedded commands in FTP requests. Each FTP command must be acknowledged before a new command is allowed. Connections sending embedded commands are dropped.
tftp	Enable TFTP application inspection. The default port is **69**.
upd	Specifies the UDP port number.

Command Modes

All **fixup protocol** commands are available in configuration mode unless otherwise specified.

The **show fixup protocol mgcp** command is available in privileged mode.

Defaults

The default ports for the PIX Firewall fixup protocols are as follows:

```
pixHA(config)# sh fix
fixup protocol dns maximum-length 512
fixup protocol ftp 21
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol http 80
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol sip 5060
fixup protocol sip udp 5060
fixup protocol skinny 2000
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol tftp 69
pixHA(config)#
```

(These are the defaults enabled on a PIX Firewall running software Version 6.3(4).)

The fixup for MGCP is disabled by default.

Usage Guidelines

The **fixup protocol** commands let you view, change, enable, or disable the use of a service or protocol through the PIX Firewall. The ports you specify are those that the PIX Firewall listens at for each respective service. You can change the port value for every service except **rsh**. The **fixup protocol** commands are always present in the configuration and are enabled by default.

The **fixup protocol** command performs the Adaptive Security Algorithm based on different port numbers other than the defaults. This command is global and changes things for both inbound and outbound connections, and cannot be restricted to any **static** command statements.

The **clear fixup** command resets the fixup configuration to its default. It does not remove the default **fixup protocol** commands.

You can disable the fixup of a protocol by removing all fixups of the protocol from the configuration using the **no fixup** command. After you remove all fixups for a protocol, the **no fixup** form of the command or the default port is stored in the configuration.

show fixup commands

The **show fixup** command displays the current fixup configuration and port values.

The **show fixup protocol** *protocol* [*protocol*] command displays the port values for the individual protocol specified.

The **show conn state** [**sip**] command displays the connection state of the designated protocol.

The **show h225**, **show h245**, and **show h323-ras** commands display connection information for troubleshooting H.323 fixup issues, and are described with the **fixup protocol h323 {h225 | ras}** commands.

The **show skinny** command assists in troubleshooting SKINNY fixup issues and is described with the **fixup protocol skinny** command.

The **show sip** command assists in troubleshooting SIP fixup issues and is described with the **fixup protocol sip udp 5060** command. The **show timeout sip** command displays the timeout value of the designated protocol.

fixup protocol ctiqbe 2748

The **fixup protocol ctiqbe 2748** command enables CTIQBE protocol inspection that supports NAT, PAT, and bi-directional NAT. This enables Cisco IP SoftPhone and other Cisco TAPI/JTAPI applications to work successfully with Cisco CallManager for call setup across the firewall.

By default, **fixup protocol ctiqbe 2748** is disabled. You enable the CTIQBE fixup as shown in the following example:

```
pixfirewall(config)# fixup protocol ctiqbe 2748

pixfirewall(config)# show fixup protocol ctiqbe
fixup protocol ctiqbe 2748
```

The **no fixup protocol ctiqbe 2748** command disables the CTIQBE fixup.

The **show ctiqbe** command displays information of CTIQBE sessions established across the PIX Firewall. Along with **debug ctiqbe** and **show local-host**, this command is used for troubleshooting CTIQBE fixup issues.

Note We recommend that you have the **pager** command configured before using the **show ctiqbe** command. If there are a lot of CTIQBE sessions and the **pager** command is not configured, it can take a while for the **show ctiqbe** command output to reach the end.

The following is sample output from the **show ctiqbe** command under the following conditions. There is only one active CTIQBE session setup across the PIX Firewall. It is established between an internal CTI device (for example, a Cisco IP SoftPhone) at local address 10.0.0.99 and an external Cisco Call Manager at 172.29.1.77, where TCP port 2748 is the Cisco CallManager. The heartbeat interval for the session is 120 seconds.

```
pixfirewall(config)# show ctiqbe

Total: 1
        LOCAL           FOREIGN         STATE   HEARTBEAT
-----------------------------------------------------------------
1       10.0.0.99/1117  172.29.1.77/2748        1       120
        ---------------------------------------------
        RTP/RTCP: PAT xlates: mapped to 172.29.1.99(1028 - 1029)
        ---------------------------------------------
        MEDIA: Device ID 27     Call ID 0
               Foreign 172.29.1.99      (1028 - 1029)
               Local   172.29.1.88      (26822 - 26823)
        ---------------------------------------------
```

The CTI device has already registered with the CallManager. The device's internal address and RTP listening port is PATed to 172.29.1.99 UDP port 1028. Its RTCP listening port is PATed to UDP 1029.

The line beginning with `RTP/RTCP: PAT xlates:` appears only if an internal CTI device has registered with an external CallManager and the CTI device's address and ports are PATed to that external interface. This line does not appear if the CallManager is located on an internal interface, or if the internal CTI device's address and ports are NATed to the same external interface that is used by the CallManager.

The output indicates a call has been established between this CTI device and another phone at 172.29.1.88. The RTP and RTCP listening ports of the other phone are UDP 26822 and 26823. The other phone locates on the same interface as the CallManager because the PIX Firewall does not maintain a CTIQBE session record associated with the second phone and CallManager. The active call leg on the CTI device side can be identified with Device ID 27 and Call ID 0.

The following is the xlate information for these CTIBQE connections:

```
pixfirewall(config)# show xlate debug
3 in use, 3 most used
Flags: D - DNS, d - dump, I - identity, i - inside, n - no random,
       o - outside, r - portmap, s - static
TCP PAT from inside:10.0.0.99/1117 to outside:172.29.1.99/1025 flags ri idle 0:00:22
timeout 0:00:30
UDP PAT from inside:10.0.0.99/16908 to outside:172.29.1.99/1028 flags ri idle 0:00:00
timeout 0:04:10
UDP PAT from inside:10.0.0.99/16909 to outside:172.29.1.99/1029 flags ri idle 0:00:23
timeout 0:04:10
```

fixup protocol dns

Use the **fixup protocol dns** command to specify the maximum DNS packet length. DNS requires application inspection so that DNS queries will not be subject to the generic UDP handling based on activity timeouts. Instead, the UDP connections associated with DNS queries and responses are torn down as soon as a reply to a DNS query has been received. This functionality is called DNS Guard.

The port assignment for the Domain Name System (DNS) is not configurable.

Set the maximum length for the DNS fixup as shown in the following example:

```
pixfirewall(config)# fixup protocol dns maximum-length 1500

pixfirewall(config)# show fixup protocol dns
fixup protocol dns maximum length 1500
```

Note The PIX Firewall drops DNS packets sent to UDP port 53 that are larger than the configured maximum length. The default value is 512 bytes. A syslog message will be generated when a DNS packet is dropped.

The **no fixup protocol dns** command disables the DNS fixup. The **clear fixup protocol dns** resets the DNS fixup to its default settings (512 byte maximum packet length).

Note If the DNS fixup is disabled, the A-record is not NATed and the DNS ID is not matched in requests and responses. By disabling the DNS fixup, the maximum length check on UDP DNS packets can be bypassed and packets greater than the maximum length configured will be permitted.

fixup protocol esp-ike

The **fixup protocol esp-ike** command enables PAT for Encapsulating Security Payload (ESP), single tunnel.

The **fixup protocol esp-ike** command is disabled by default. If a **fixup protocol esp-ike** command is issued**,** the fixup is turned on, and the firewall preserves the source port of the Internet Key Exchange (IKE) and creates a PAT translation for ESP traffic. Additionally, if the **esp-ike** fixup is on, ISAKMP cannot be turned on any interface.

fixup protocol ftp

Use the **fixup protocol ftp** command to specify the listening port or ports for the File Transfer Protocol (FTP). The following list describes the features and usage of this command:

- The PIX Firewall listens to port 21 for FTP by default.
- Mutliple ports can be specified.
- Only specify the port for the FTP control connection and not the data connection. The PIX Firewall stateful inspection will dynamically prepare the data connection as necessary. For example, the following is incorrect:

  INCORRECT

  ```
  fixup protocol ftp 21
  fixup protocol ftp 20
  ```

  and is the following is correct:

  CORRECT

  ```
  fixup protocol ftp 21
  ```

- Use caution when moving FTP to a higher port. For example, if you set the FTP port to 2021 by entering **fixup protocol ftp 2021** all connections that initiate to port 2021 will have their data payload interpreted as FTP commands.

The following is an example of a **fixup protocol ftp** command configuration that uses multiple FTP fixups:

```
:
: For a PIX Firewall with two interfaces
:
ip address outside 192.168.1.1 255.255.255.0
ip address inside 10.1.1.1 255.255.255.0
:
: There is an inside host 10.1.1.15 that will be
: exported as 192.168.1.15.  This host runs the FTP
: services at port 21 and 1021
:
static (inside, outside) 192.168.1.15 10.1.1.15
:
: Construct an access list to permit inbound FTP traffic to
: port 21 and 1021
:
access-list outside permit tcp any host 192.168.1.15 eq ftp
access-list outside permit tcp any host 192.168.1.15 eq 1021
access-group outside in interface outside
:
: Specify that traffic to port 21 and 1021 are FTP traffic
:
fixup protocol ftp 21
fixup protocol ftp 1021
```

If you disable FTP fixups with the **no fixup protocol ftp** command, outbound users can start connections only in passive mode, and all inbound FTP is disabled.

The **strict** option in the **fixup protocol ftp** command performs two seperate functions:

- The **strict** option prevents web browsers from sending embedded commands in FTP requests. Each FTP command must be acknowledged before a new command is allowed. Connections sending embedded commands are dropped.The **strict** option only lets an FTP server generate the 227 command and only lets an FTP client generate the PORT command. The 227 and PORT commands are checked to ensure they do not appear in an error string.
- The **strict** option also prevents the PIX from opening up return connections based solely on the information sent in the PORT command. The **strict** option enables the PIX to make sure a successful reply is sent from the server in addition to the PORT command before opening the connection. If an error is seen, the PORT command is ignored by the PIX and the connection is never established. This keeps the PIX from opening data connections for communication that will never occur.

fixup protocol h323 {h225 | ras}

The **fixup protocol h323** {**h225 | ras**} command provides support for H.323 compliant applications such as Cisco CallManager and VocalTec Gatekeeper. H.323 is a suite of protocols defined by the International Telecommunication Union (ITU) for multimedia conferences over LANs.

PIX Firewall software Version 5.3 and higher supports H.323 v2 with Fast Connect or Fast Start Procedure for faster call setup and H.245 tunneling for resource conservation, call synchronization, and reduced set up time.

PIX Firewall software Versions 6.2 and higher support PAT for H.323. When upgrading from any pre-PIX Firewall software Version 6.2 release, the following will be added to the configuration:

```
fixup protocol h323 ras 1718-1719
```

Additionally, **fixup protocol h323** *port* becomes **fixup protocol h323 h225** *port*. You can disable H.225 signaling or RAS fixup (or both) with the **no fixup protocol h323** {**h225 | ras**} *port* [**-***port*] command.

PIX Firewall software Version 6.3 and higher supports H.323 v3 and v4 messages as well as the H.323 v3 feature Multiple Calls on One Call Signaling Channel.

The **show h225**, **show h245**, and **show h323-ras** commands display connection information for troubleshooting H.323 fixup issues.

Note Before using the **show h225**, **show h245**, or **show h323-ras** commands, we recommend that you configure the **pager** command. If there are a lot of session records and the **pager** command is not configured, it may take a while for the **show** output to reach its end. If there is an abnormally large number of connections, check that the sessions are timing out based on the default timeout values or the values set by you. If they are not, then there is a problem that needs to be investigated.

The **show h225** command displays information for H.225 sessions established across the PIX Firewall. Along with the **debug h323 h225 event**, **debug h323 h245 event**, and **show local-host** commands, this command is used for troubleshooting H.323 fixup issues.

The following is sample output from the **show h225** command:

```
pixfirewall(config)# show h225
Total H.323 Calls: 1
1 Concurrent Call(s) for
    Local:   10.130.56.3/1040   Foreign: 172.30.254.203/1720
    1. CRV 9861
    Local:   10.130.56.3/1040   Foreign: 172.30.254.203/1720
0 Concurrent Call(s) for
    Local:   10.130.56.4/1050   Foreign: 172.30.254.205/1720
```

This output indicates that there is currently 1 active H.323 call going through the PIX Firewall between the local endpoint 10.130.56.3 and foreign host 172.30.254.203, and for these particular endpoints, there is 1 concurrent call between them, with a CRV (Call Reference Value) for that call of 9861.

For the local endpoint 10.130.56.4 and foreign host 172.30.254.205, there are 0 concurrent Calls. This means that there is no active call between the endpoints even though the H.225 session still exists. This could happen if, at the time of the **show h225** command, the call has already ended but the H.225 session has not yet been deleted. Alternately, it could mean that the two endpoints still have a TCP connection opened between them because they set "maintainConnection" to TRUE, so the session is kept open until they set it to FALSE again, or until the session times out based on the H.225 timeout value in your configuration.

The **show h245** command displays information for H.245 sessions established across the PIX Firewall by endpoints using slow start. (Slow start is when the two endpoints of a call open another TCP control channel for H.245. Fast start is where the H.245 messages are exchanged as part of the H.225 messages on the H.225 control channel.) Along with the **debug h323 h245 event**, **debug h323 h225 event**, and **show local-host** commands, this command is used for troubleshooting H.323 fixup issues.

The following is sample output from the **show h245** command:

```
pixfirewall(config)# show h245
Total: 1
        LOCAL           TPKT    FOREIGN         TPKT
1       10.130.56.3/1041        0       172.30.254.203/1245    0
        MEDIA: LCN 258 Foreign 172.30.254.203 RTP 49608 RTCP 49609
                       Local   10.130.56.3 RTP 49608 RTCP 49609
        MEDIA: LCN 259 Foreign 172.30.254.203 RTP 49606 RTCP 49607
                       Local   10.130.56.3 RTP 49606 RTCP 49607
```

There is currently one H.245 control session active across the PIX Firewall. The local endpoint is 10.130.56.3, and we are expecting the next packet from this endpoint to have a TPKT header since the TPKT value is 0. (The TKTP header is a 4-byte header preceding each H.225/H.245 message. It gives

the length of the message, including the 4-byte header.) The foreign host endpoint is 172.30.254.203, and we are expecting the next packet from this endpoint to have a TPKT header since the TPKT value is 0.

The media negotiated between these endpoints have a LCN (logical channel number) of 258 with the foreign RTP IP address/port pair of 172.30.254.203/49608 and a RTCP IP address/port of 172.30.254.203/49609 with a local RTP IP address/port pair of 10.130.56.3/49608 and a RTCP port of 49609.

The second LCN of 259 has a foreign RTP IP address/port pair of 172.30.254.203/49606 and a RTCP IP address/port pair of 172.30.254.203/49607 with a local RTP IP address/port pair of 10.130.56.3/49606 and RTCP port of 49607.

The **show h323-ras** command displays information for H.323 RAS sessions established across the PIX Firewall between a gatekeeper and its H.323 endpoint. Along with the **debug h323 ras event** and **show local-host** commands, this command is used for troubleshooting H.323 RAS fixup issues.

The following is sample output from the **show h323-ras** command:

```
pixfirewall(config)# show h323-ras
Total: 1
        GK                      Caller
        172.30.254.214 10.130.56.14
```

This output shows that there is one active registration between the gatekeeper 172.30.254.214 and its client 10.130.56.14.

fixup protocol http

The **fixup protocol http** command sets the port for Hypertext Transfer Protocol (HTTP) traffic application inspection. The default port for HTTP is 80.

Use the port option to change the default port assignments from 80. Use the *port-port* option to apply HTTP application inspection to a range of port numbers.

Note The **no fixup protocol http** command statement also disables the **filter url** command.

HTTP inspection performs several functions:

- URL logging of GET messages
- URL screening through N2H2 or Websense
- Java and ActiveX filtering

The latter two features must be configured in conjuction with the **filter** command.

fixup protocol icmp error

The **fixup protocol icmp error** command enables NAT of ICMP error messages. This creates translations for intermediate hops based on the static or network address translation configuration on the firewall.

The **no fixup protocol icmp error** disables the creation of a translation (xlate) for the intermediate nodes which generate ICMP error messages.

By default **fixup protocol icmp error** is disabled.

fixup protocol ils

The **fixup protocol ils** command provides NAT support for Microsoft NetMeeting, SiteServer, and Active Directory products that use LightWeight Directory Access Protocol (LDAP) to exchange directory information with an for Internet Locator Service (ILS) server.

By default, **fixup protocol ils** is disabled. You enable the ILS fixup as shown in the following example:

```
pixfirewall(config)# fixup protocol ils
```

The **no fixup protocol ils** command disables the ILS fixup.

fixup protocol mgcp

If a user wishes to use MGCP, they will usually need to configure at least two **fixup protocol** commands: one for the port on which the gateway receives commands, and one for the port on which the Call Agent receives commands.

Normally, a Call Agent sends commands to the default MGCP port for gateways, 2427, and a gateway sends commands to the default MGCP port for Call Agents, 2727.

The following example adds fixup support for Call Agents and gateways that use the default ports:

```
fixup protocol mgcp 2427
fixup protocol mgcp 2727
```

The **no fixup protocol mgcp** command removes the MGCP fixup configuration.

The **show fixup protocol mgcp** command displays the configured MGCP fixups. Please refer to the **mgcp** command for information on the **show mgcp** command.

fixup protocol pptp

The **fixup protocol pptp** [**1723**] command inspects PPTP protocol packets and dynamically creates the GRE connections and xlates necessary to permit PPTP traffic.

Specifically, the firewall inspects the PPTP version announcements and the outgoing call request/response sequence. Only PPTP Version 1, as defined in RFC 2637, is inspected. Further inspection on the TCP control channel is disabled if the version announced by either side is not Version 1. In addition, the outgoing-call request and reply sequence are tracked. Connections and/or xlates are dynamic allocated as necessary to permit subsequent secondary GRE data traffic.

The **fixup protocol pptp 1723** command is disabled by default. Enter the following command to enable the PPTP fixup:

```
pixfirewall(config)# fixup protocol pptp 1723
pixfirewall(config)# show fixup
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol ils 389
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
fixup protocol pptp 1723
fixup protocol sip udp 5060
fixup protocol tftp 69
```

The PPTP fixup must be enabled for PPTP traffic to be translated by PAT. Additionally, PAT is only performed for a modified version of GRE (RFC2637) and only if it is negotiated over the PPTP TCP control channel. PAT is not performed for the unmodified version of GRE (RFC 1701 and RFC 1702).

fixup protocol rtsp

The **fixup protocol rtsp** command lets PIX Firewall pass Real Time Streaming Protocol (RTSP) packets. RTSP is used by RealAudio, RealNetworks, Apple QuickTime 4, RealPlayer, and Cisco IP/TV connections.

If you are using Cisco IP/TV, use RTSP TCP port 554 and TCP 8554:

```
fixup protocol rtsp 554
fixup protocol rtsp 8554
```

The following restrictions apply to the **fixup protocol rtsp** command:

1. This PIX Firewall will not fix RTSP messages passing through UDP ports.
2. PAT is not supported with the **fixup protocol rtsp** command.
3. PIX Firewall does not have the ability to recognize HTTP cloaking where RTSP messages are hidden in the HTTP messages.
4. PIX Firewall cannot perform NAT on RTSP messages because the embedded IP addresses are contained in the SDP files as part of HTTP or RTSP messages. Packets could be fragmented and PIX Firewall cannot perform NAT on fragmented packets.
5. With Cisco IP/TV, the number of NATs the PIX Firewall performs on the SDP part of the message is proportional to the number of program listings in the Content Manager (each program listing can have at least six embedded IP addresses).
6. You can configure NAT for Apple QuickTime 4 or RealPlayer. Cisco IP/TV only works with NAT if the Viewer and Content Manager are on the outside network and the server is on the inside network.
7. When using RealPlayer, it is important to properly configure transport mode. For the PIX Firewall, add an **access-list** command statement from the server to the client or vice versa. For RealPlayer, change transport mode by clicking **Options>Preferences>Transport>RTSP Settings**.

   If using TCP mode on the RealPlayer, select the **Use TCP to Connect to Server** and **Attempt to use TCP for all content** check boxes. On the PIX Firewall, there is no need to configure the fixup.

   If using UDP mode on the RealPlayer, select the **Use TCP to Connect to Server** and **Attempt to use UDP for static content** check boxes, and for live content not available via Multicast. On the PIX Firewall, add a **fixup protocol rtsp** *port* command statement.

fixup protocol sip

Use the **fixup protocol sip** [*port*[**-***port*]] command to enable SIP-over-TCP application inspection, or the **fixup protocol sip udp 5060** command to enable SIP-over-UDP application inspection. If either SIP fixup method is enabled, SIP packets are inspected and then NAT is provided for the appropriate IP addresses. The SIP fixups are enabled by default on TCP or UDP port 5060, respectively. However, only the TCP SIP fixup port is configurable in PIX Firewall software Version 6.3. You cannot change ports on the firewall for the SIP-over-UDP fixup, but you can disable the SIP-over-UDP fixup with the **no fixup protocol sip udp 5060** command.

Session Initiation Protocol (SIP), as defined by the Internet Engineering Task Force (IETF), enables call handling sessions and two-party audio conferences (calls). SIP works with Session Description Protocol (SDP) for call signaling. SDP specifies the ports for the media stream. Using SIP, the PIX Firewall can support any SIP Voice over IP (VoIP) gateway or VoIP proxy server. SIP and SDP are defined in the following RFCs:

- SIP: Session Initiation Protocol, RFC 2543
- SDP: Session Description Protocol, RFC 2327

To support SIP, the following must be inspected: calls through the PIX Firewall, signaling messages for the media connection addresses, media ports, and embryonic connections for the media. This is because while the signaling is sent over a well known destination port (UDP/TCP 5060), the media streams are dynamically allocated and because SIP is a text-based protocol that contains IP addresses throughout the text.

PIX Firewall software Version 6.2 and higher supports PAT for SIP. In PIX Firewall software Version 6.3 and later, you can disable the SIP fixup for both UDP and TCP signaling with the commands **no fixup protocol sip udp 5060** and **no fixup protocol sip** [*port*[**-***port*] respectively.

For additional information about the SIP protocol see RFC 2543. For additional information about the Session Description Protocol (SDP), see RFC 2327.

The **show sip** command displays information for SIP sessions established across the PIX Firewall. Along with the **debug sip** and **show local-host** commands, this command is used for troubleshooting SIP fixup issues.

Note We recommend that you configure the **pager** command before using the **show sip** command. If there are a lot of SIP session records and the **pager** command is not configured, it will take a while for the **show sip** command output to reach its end.

The following is sample output from the **show sip** command:

```
pixfirewall(config)# show sip
Total: 2
call-id c3943000-960ca-2e43-228f@10.130.56.44
    state Call init, idle 0:00:01
call-id c3943000-860ca-7e1f-11f7@10.130.56.45
    state Active, idle 0:00:06
```

This sample shows two active SIP sessions on the PIX Firewall (as shown in the `Total` field). Each `call-id` represents a call.

The first session, with the `call-id c3943000-960ca-2e43-228f@10.130.56.44,` is in the state `Call Init`, which means the session is still in call setup. Call setup is not complete until a final response to the call has been received. For instance, the caller has already sent the INVITE, and maybe received a 100 Response, but has not yet seen the 200 OK, so the call setup is not complete yet. Any non-1xx response message is considered a final response. This session has been idle for 1 second.

The second session is in the state `Active`, in which call setup is complete and the endpoints are exchanging media. This session has been idle for 6 seconds.

fixup protocol skinny

Skinny Client Control Protocol (SCCP or “skinny”) protocol supports IP telephony and can coexist in an H.323 environment. An application layer ensures that all SCCP signaling and media packets can traverse the PIX Firewall and interoperate with H.323 terminals. The skinny fixup supports both NAT and PAT configurations.

Note If the address of an internal Cisco CallManager is configured for NAT or PAT to a different IP address or port, registrations for external Cisco IP Phones will fail because the PIX Firewall currently does not support NAT or PAT for the file content transferred via TFTP. Although the PIX Firewall does support NAT of TFTP messages, and opens a pinhole for the TFTP file to traverse the firewall, the PIX Firewall cannot translate the Cisco CallManager IP address and port embedded in the Cisco IP Phone's configuration files that are being transferred using TFTP during phone registration.

If skinny messages are fragmented, then the firewall does not recognize or inspect them. Skinny message fragmentation can occur when a call is established that includes a conference bridge. The firewall tracks the skinny protocol to open conduits for RTP traffic to flow through, however, with the skinny messages fragmented, the firewall cannot correctly set up this conduit.

The **show skinny** command displays information of Skinny (SCCP) sessions established across the PIX Firewall. Along with **debug skinny** and **show local-host**, this command is used for troubleshooting Skinny fixup issues.

Note We recommend that you have the **pager** command configured before using the **show skinny** command. If there are a lot of Skinny sessions and the **pager** command is not configured, it can take a while for the **show skinny** command output to reach the end.

The following is sample output from the **show skinny** command under the following conditions. There are two active Skinny sessions set up across the PIX Firewall. The first one is established between an internal Cisco IP Phone at local address 10.0.0.11 and an external Cisco CallManager at 172.18.1.33. TCP port 2000 is the CallManager. The second one is established between another internal Cisco IP Phone at local address 10.0.0.22 and the same Cisco CallManager.

```
pixfirewall(config)# show skinny
        LOCAL                   FOREIGN                 STATE
----------------------------------------------------------------
1       10.0.0.11/52238         172.18.1.33/2000                1
  MEDIA 10.0.0.11/22948         172.18.1.22/20798
2       10.0.0.22/52232         172.18.1.33/2000                1
  MEDIA 10.0.0.22/20798         172.18.1.11/22948
```

The output indicates a call has been established between both internal Cisco IP Phones. The RTP listening ports of the first and second phones are UDP 22948 and 20798 respectively.

The following is the xlate information for these Skinny connections:

```
pixfirewall(config)# show xlate debug
2 in use, 2 most used
Flags: D - DNS, d - dump, I - identity, i - inside, n - no random,
       o - outside, r - portmap, s - static
NAT from inside:10.0.0.11 to outside:172.18.1.11 flags si idle 0:00:16 timeout 0:05:00
NAT from inside:10.0.0.22 to outside:172.18.1.22 flags si idle 0:00:14 timeout 0:05:00
```

fixup protocol smtp

The **fixup protocol smtp** command enables the Mail Guard feature, which only lets mail servers receive the RFC 821, section 4.5.1, commands of HELO, MAIL, RCPT, DATA, RSET, NOOP, and QUIT. All other commands are translated into X's which are rejected by the internal server. This results in a message such as “500 Command unknown: 'XXX'.” Incomplete commands are discarded.

Note During an interactive SMTP session, various SMTP security rules may reject or deadlock your Telnet session. These rules include the following: SMTP commands must be at least four characters in length; must be terminated with carriage return and line feed; and must wait for a response before issuing the next reply.

As of PIX Firewall software Version 5.1 and higher, the **fixup protocol smtp** command changes the characters in the SMTP banner to asterisks except for the "2", "0", "0 " characters. Carriage return (CR) and linefeed (LF) characters are ignored.

In PIX Firewall software Version 4.4, all characters in the SMTP banner are converted to asterisks.

fixup protocol snmp

This snmp fixup command **fixup protocol snmp 161-162** is disabled by default. This command provides the ability to configure a drop of SNMP packets based on protocol version.

The **no fixup protocol snmp** command removes the SNMP fixup configuration.

Fixup can be enabled or disabled with the following command:

[no] fixup protocol snmp 161-162

Note Existing connections will retain present fixup configurations from their initial creation.

So, if you toggle the configuration, you need to either:

- Wait for the connections to time out
- Manually clear the connections

Use **clear xlate** or **clear local** to clear connections for the fixup configuration to take effect.

fixup protocol sqlnet

PIX Firewall uses port 1521 for SQL*Net. This is the default port used by Oracle for SQL*Net; however, this value does not agree with IANA port assignments.

fixup protocol tftp

PIX Firewall Version 6.3(2) introduced application inspection for Trivial File Transfer Protocol (TFTP). The default port is 69. Use the *port-port* option to apply TFTP application inspection to a range of port numbers.

The PIX Firewall inspects TFTP traffic and dynamically creates connections and translations, if necessary, to permit file transfer between a TFTP client and server with the **fixup protocol tftp** command. Specifically, the fixup inspects TFTP read request (RRQ), write request (WRQ), and error notification (ERROR).

A dynamic secondary channel and a PAT translation, if necessary, are allocated on a reception of a valid read (RRQ) or write (WRQ) request. This secondary channel is subsequently used by TFTP for file transfer or error notification.

Only the TFTP server can initiate traffic over the secondary channel, and at most one incomplete secondary channel can exist between the TFTP client and server. An error notification from the server closes the secondary channel.

The show **fixup protocol tftp** command displays the ports on which TFTP is inspected.

```
pixdoc515(config)# show fixup protocol tftp
fixup protocol tftp 69
```

Examples

The following example enables access to an inside server running Mail Guard:

```
static (inside,outside) 209.165.201.1 192.168.42.1 netmask 255.255.255.255
access-list acl_out permit tcp any host 209.165.201.1 eq smtp
access-group acl_out in interface outside
fixup protocol smtp 25
```

The following example shows the commands to disable Mail Guard:

```
static (dmz1,outside) 209.165.201.1 10.1.1.1 netmask 255.255.255.255
access-list acl_out permit tcp any host 209.165.201.1 eq smtp
access-group acl_out in interface outside
no fixup protocol smtp 25
```

In this example, the **static** command sets up a global address to permit outside hosts access to the 10.1.1.1 mail server host on the dmz1 interface. (The MX record for DNS needs to point to the 209.165.201.1 address so that mail is sent to this address.) The **access-list** command lets any outside users access the global address through the SMTP port (25). The **no fixup protocol** command disables the Mail Guard feature.

The following example shows how to enable the MGCP fixup on your firewall:

```
pixfirewall(config)# fixup protocol mgcp 2427
pixfirewall(config)# fixup protocol mgcp 2727
pixfirewall(config)# show running-config
: Saved
:
PIX Version 6.3
interface ethernet0 auto
interface ethernet1 auto
interface ethernet2 auto shutdown
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 intf2 security10
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname pixfirewall
domain-name cisco.com
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol ils 389
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
fixup protocol mgcp 2427
fixup protocol mgcp 2727
fixup protocol sip udp 5060
fixup protocol tftp 69
names
access-list 101 permit tcp any host 10.1.1.3 eq www
access-list 101 permit tcp any host 10.1.1.3 eq smtp
pager lines 24
mtu outside 1500
mtu inside 1500
mtu intf2 1500
ip address outside 172.23.59.232 255.255.0.0
ip address inside 10.1.1.1 255.255.255.0
ip address intf2 127.0.0.1 255.255.255.255
ip audit info action alarm
```

```
ip audit attack action alarm
pdm history enable
arp timeout 14400
global (outside) 1 interface
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
routing interface inside
route outside 0.0.0.0 0.0.0.0 172.23.59.225 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h225 1:00:00
timeout h323 0:05:00 mgcp 0:05:00 sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
aaa-server LOCAL protocol local
http server enable
http 10.1.1.2 255.255.255.255 inside
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
floodguard enable
telnet timeout 5
ssh timeout 5
console timeout 0
dhcprelay server 10.1.1.1 outside
terminal width 80
Cryptochecksum:00000000000000000000000000000000
: end
```

The following example shows how to remove the MGCP fixup from your configuration:

```
pixfirewall(config)# show fixup protocol mgcp
fixup protocol mgcp 2427
fixup protocol mgcp 2727
pixfirewall(config)# no fixup protocol mgcp
pixfirewall(config)#
```

Related Commands

debug	Displays debug information for Media Gateway Control Protocol (MGCP) traffic.
mgcp	Configures additional support for the Media Gateway Control Protocol fixup (packet application inspection) and is used with the **fixup protocol mgcp** command.
show conn	Displays all active connections. There is an MGCP **show conn** option and connection flag, “g”.
timeout	Sets the maximum idle time duration. (There is an MGCP timeout option.)

flashfs

Clear, display, or downgrade filesystem information.

flashfs downgrade {**4.x** | **5.0** | **5.1**}

clear flashfs

show flashfs

Syntax Description

downgrade 4.x	Clear the filesystem information from Flash memory before downgrading to PIX Firewall software Version 4.0, 4.1, 4.2, 4.3, or 4.4.
downgrade 5.0 | 5.1	Write the filesystem to Flash memory before downgrading to the appropriate PIX Firewall software Version 5.0 or higher.

Command Modes

Configuration mode.

Usage Guidelines

The **clear flashfs** and the **flashfs downgrade 4.x** commands clear the filesystem part of Flash memory in the PIX Firewall. Versions 4.*n* cannot use the information in the filesystem; it needs to be cleared to let the earlier version operate correctly.

The **flashfs downgrade 5.x** command reorganizes the filesystem part of Flash memory so that information stored in the filesystem can be accessed by the earlier version. The PIX Firewall maintains a filesystem in Flash memory to store system information, IPSec private keys, certificates, and CRLs. It is crucial that you clear or reformat the filesystem before downgrading to a previous PIX Firewall version. Otherwise, your filesystem will get out of sync with the actual contents of the Flash memory and cause problems when the unit is later upgraded.

Note When downgrading to PIX Firewall Versions 5.0 or 5.1, which support a maximum 4 MB of Flash memory, configuration files larger than 4 MB will be truncated and some configuration information will be lost.

You only need to use the **flashfs downgrade 5.x** command if your PIX Firewall has 16 MB of Flash memory, if you have IPSec private keys, certificates, or CRLs stored in Flash memory, and you used the **ca save all** command to save these items in Flash memory. The **flashfs downgrade 5.x** command fails if the filesystem indicates that any part of the image, configuration, or private data in the Flash memory device is unusable.

The **clear flashfs** and **flashfs downgrade** commands do not affect the configuration stored in Flash memory.

The **clear flashfs** command is the same as the **flashfs downgrade 4.x** command.

The **show flashfs** command displays the size in bytes of each filesystem sector and the current state of the filesystem. The data in each sector is as follows:

- file 0—PIX Firewall binary image, where the .bin file is stored.
- file 1—PIX Firewall configuration data that you can view with the **show config** command.
- file 2—PIX Firewall datafile that stores IPSec key and certificate information.
- file 3—**flashfs downgrade** information for the **show flashfs** command.
- file 4—The compressed PIX Firewall image size in Flash memory.

Examples

The following is sample output from the **show flashfs** command:

```
pixfirewall(config)# show flashfs
flash file system:  version:2  magic:0x12345679
```

```
  file 0: origin:       0 length:1511480
  file 1: origin: 2883584 length:3264
  file 2: origin:       0 length:0
  file 3: origin: 3014656 length:4444164
  file 4: origin: 8257536 length:280
```

Use the following command to write the filesystem to Flash memory before downgrading to a lower version of software:

```
pixfirewall(config)# flashfs downgrade 5.3
```

The following commands display the filesystem sector sizes:

```
pixfirewall(config)# show flashfs
flash file system:  version:1  magic:0x12345679
    file 0: origin:       0 length:1794104
    file 1: origin: 2095104 length:1496
    file 2: origin:       0 length:0
    file 3: origin: 2096640 length:140
    file 4: origin: 8257536 length:280

pixfirewall(config)# flashfs downgrade 5.3
pixfirewall(config)# show flashfs
flash file system:  version:0  magic:0x0
    file 0: origin:       0 length:0
    file 1: origin:       0 length:0
    file 2: origin:       0 length:0
    file 3: origin:       0 length:0
    file 4: origin: 8257536 length:280
```

The origin values are integer multiples of the underlying filesystem sector size.

floodguard

Enable or disable Flood Guard to protect against flood attacks.

floodguard enable

floodguard disable

clear floodguard

show floodguard

Syntax Description

enable	Enable Flood Guard.
disable	Disable Flood Guard.

Command Modes

Configuration mode.

Usage Guidelines

The **floodguard** command lets you reclaim PIX Firewall resources if the user authentication (uauth) subsystem runs out of resources. If an inbound or outbound uauth connection is being attacked or overused, the PIX Firewall will actively reclaim TCP user resources.

When the resources deplete, the PIX Firewall lists messages about it being out of resources or out of tcpusers.

If the PIX Firewall uauth subsystem is depleted, TCP user resources in different states are reclaimed depending on urgency in the following order:

1. Timewait
2. LastAck
3. FinWait
4. Embryonic
5. Idle

The **floodguard** command is enabled by default.

Examples

The following example enables the **floodguard** command and lists the **floodguard** command statement in the configuration:

```
floodguard enable
show floodguard
floodguard enable
```

fragment

The **fragment** command provides additional management of packet fragmentation and improves compatibility with NFS.

fragment size *database-limit* [*interface*]

fragment chain *chain-limit* **[***interface*]

fragment timeout *seconds* [*interface*]

clear fragment

show fragment [*interface*]

Syntax Description

chain	Specifies the maximum number of packets into which a full IP packet can be fragmented. The default is 24.
chain-limit	The default is 24. The maximum is 8200.
clear	Resets the fragment databases and defaults. All fragments currently waiting for reassembly are discarded and the **size**, **chain**, and **timeout** options are reset to their default values.
database-limit	The default is 200. The maximum is 1,000,000 or the total number of blocks.
interface	The PIX Firewall interface. If not specified, the command will apply to all interfaces.
seconds	The default is 5 seconds. The maximum is 30 seconds.
show	• Displays the state of the fragment database: • Size—Maximum packets set by the **size** option. • Chain—Maximum fragments for a single packet set by the **chain** option. • Timeout—Maximum seconds set by the **timeout** option. • Queue—Number of packets currently awaiting reassembly. • Assemble—Number of packets successfully reassembled. • Fail—Number of packets which failed to be reassembled. • Overflow—Number of packets which overflowed the fragment database.
size	Sets the maximum number of packets in the fragment database. The default is 200.
timeout	Specifies the maximum number of seconds that a packet fragment will wait to be reassembled after the first fragment is received before being discarded. The default is 5 seconds.

Command Modes

Configuration mode.

Usage Guidelines

By default the PIX Firewall accepts up to 24 fragments to reconstruct a full IP packet. Based on your network security policy, you should consider configuring the PIX Firewall to prevent fragmented packets from traversing the firewall by entering the **fragment chain 1** *interface* command on each interface. Setting the limit to 1 means that all packets must be whole; that is, unfragmented.

If a large percentage of the network traffic through the PIX Firewall is NFS, additional tuning may be necessary to avoid database overflow. See system log message 209003 for additional information.

In an environment where the MTU between the NFS server and client is small, such as a WAN interface, the **chain** option may require additional tuning. In this case, NFS over TCP is highly recommended to improve efficiency.

Setting the *database-limit* of the **size** option to a large value can make the PIX Firewall more vulnerable to a DoS attack by fragment flooding. Do not set the *database-limit* equal to or greater than the total number of blocks in the 1550 or 16384 pool. See the **show block** command for more details. The default values will limit DoS due to fragment flooding to that interface only.

The **show fragment** [*interface*] command displays the states of the fragment databases. If the interface name is specified, only displays information for the database residing at the specified interface.

Examples

For example, to prevent fragmented packets on the outside and inside interfaces enter the following commands:

```
pixfirewall(config)# fragment chain 1 outside
pixfirewall(config)# fragment chain 1 inside
```

Continue entering the **fragment chain 1** *interface* command for each additional interface on which you want to prevent fragmented packets.

The following example configures the outside fragment database to limit a maximum size of 2000, a maximum chain length of 45, and a wait time of 10 seconds:

```
pixfirewall(config)#
pixfirewall(config)# fragment outside size 2000
pixfirewall(config)# fragment chain 45 outside
pixfirewall(config)# fragment outside timeout 10
pixfirewall(config)#
```

The **clear fragment** command resets the fragment databases. Specifically, all fragments awaiting re-assembly are discarded. In addition, the size is reset to 200; the chain limit is reset to 24; and the timeout is reset to 5 seconds.

The **show fragment** command display the states of the fragment databases. If the interface name is specified, only the database residing at the specified interface is displayed.

```
pixfirewall(config)# show fragment outside
Interface:outside
Size:2000, Chain:45, Timeout:10
Queue:1060, Assemble:809, Fail:0, Overflow:0
```

The preceding example shows that the "outside" fragment database has the following:

- A database size limit of 2000 packets.
- The chain length limit of 45 fragments.
- A timeout of ten seconds.
- 1060 packets is currently awaiting re-assembly.
- 809 packets has been fully reassembled.

- No failure.
- No overflow.

This fragment database is under heavy usage.

The PIX Firewall also includes FragGuard for additional IP fragmentation protection. For more information refer to the *Cisco PIX Firewall and VPN Configuration Guide*.

CHAPTER 6

G through L Commands

global

Create or delete entries from a pool of global addresses.

[no] global [(*if_name*)] *nat_id* {*global_ip* [-*global_ip*] [**netmask** *global_mask*]} | **interface**

clear global

show global

Syntax Description

clear	Removes **global** command statements from the configuration.
global_ip	One or more global IP addresses that the PIX Firewall shares among its connections. If the external network is connected to the Internet, each global IP address must be registered with the Network Information Center (NIC). You can specify a range of IP addresses by separating the addresses with a dash (-). You can create a Port Address Translation (PAT) **global** command statement by specifying a single IP address. You can have more than one PAT **global** command statement per interface. A PAT can support up to 65,535 xlate objects.
global_mask	The network mask for *global_ip*. If subnetting is in effect, use the subnet mask; for example, 255.255.255.128. If you specify an address range that overlaps subnets, **global** will not use the broadcast or network addresses in the pool of global addresses. For example, if you use 255.255.255.224 and an address range of 209.165.201.1-209.165.201.30, the 209.165.201.31 broadcast address and the 209.165.201.0 network address will not be included in the pool of global addresses.
if_name	The external network where you use these global addresses.
interface	Specifies PAT using the IP address at the interface.
nat_id	A positive number shared with the **nat** command that groups the **nat** and **global** command statements together. The valid ID numbers can be any positive number up to 2,147,483,647.
netmask	Reserved word that prefaces the network *global_mask* variable.

Command Modes

Configuration mode.

Usage Guidelines

The **global** command defines a pool of global addresses. The global addresses in the pool provide an IP address for each outbound connection, and for those inbound connections resulting from outbound connections. Ensure that associated **nat** and **global** command statements have the same *nat_id*.

Use caution with names that contain a "-" (dash) character because the **global** command interprets the last (or only) "-" character in the name as a range specifier instead of as part of the name. For example, the **global** command treats the name "`host-net2`" as a range from "`host`" to "`net2`". If the name is "`host-net2-section3`" then it is interpreted as a range from "`host-net2`" to "`section3`".

The following command form is used for Port Address Translation (PAT) only:
global [(*if_name*)] *nat_id* {{*global_ip*} [**netmask** *global_mask*] | **interface**}

After changing or removing a **global** command statement, use the **clear xlate** command.

Use the **no global** command to remove access to a *nat_id*, or to a Port Address Translation (PAT) address, or address range within a *nat_id*.

The **show global** command displays the **global** command statements in the configuration.

PAT

You can enable the Port Address Translation (PAT) feature by entering a single IP address with the **global** command. PAT lets multiple outbound sessions appear to originate from a single IP address. With PAT enabled, the PIX Firewall chooses a unique port number from the PAT IP address for each outbound xlate (translation slot). This feature is valuable when an Internet service provider cannot allocate enough unique IP addresses for your outbound connections. An IP address you specify for a PAT cannot be used in another global address pool.

When a PAT augments a pool of global addresses, first the addresses from the global pool are used, then the next connection is taken from the PAT address. If a global pool address is available, the next connection takes that address. The global pool addresses always come first, before a PAT address is used. Augment a pool of global addresses with a PAT by using the same *nat_id* in the **global** command statements that create the global pools and the PAT.

For example:

```
global (outside) 1 209.165.201.1-209.165.201.10 netmask 255.255.255.224
global (outside) 1 209.165.201.22 netmask 255.255.255.224
```

PAT does not work with H.323 applications and caching nameservers. Do not use a PAT when multimedia applications need to be run through the PIX Firewall. Multimedia applications can conflict with port mappings provided by PAT.

The firewall does not PAT all ICMP message types; it only PATs ICMP echo and echo-reply packets (types 8 and 0). Specifically, only ICMP echo or echo-reply packets create a PAT xlate. So, when the other ICMP messages types are dropped, syslog message 305006 (on the PIX Firewall) is generated.

PAT does not work with the **established** command. PAT works with DNS, FTP and passive FTP, HTTP, email, RPC, rshell, Telnet, URL filtering, and outbound traceroute.

However, for use with passive FTP, use the **fixup protocol ftp strict** command statement with an **access-list** command statement to permit outbound FTP traffic, as shown in the following example:

```
fixup protocol ftp strict ftp
access-list acl_in permit tcp any any eq ftp
access-group acl_in in interface inside
nat (inside) 1 0 0
global (outside) 1 209.165.201.5 netmask 255.255.255.224
```

To specify PAT using the IP address of an interface, specify the **interface** keyword in the **global** [(*int_name*)] *nat_id address* | **interface** command.

The following example enables PAT using the IP address at the outside interface in global configuration mode:

```
ip address outside 192.150.49.1
nat (inside) 1 0 0
global (outside) 1 interface
```

The interface IP address used for PAT is the address associated with the interface when the xlate (translation slot) is created. This is important for configuring DHCP, allowing for the DHCP retrieved address to be used for PAT.

When PAT is enabled on an interface, there should be no loss of TCP, UDP, and ICMP services. These services allow for termination at the PIX Firewall unit's outside interface.

To track usage among different subnets, you can specify multiple PATs using the following supported configurations:

The following example maps hosts on the internal network 10.1.0.0/24 to global address 192.168.1.1 and hosts on the internal network 10.1.1.1/24 to global address 209.165.200.225 in global configuration mode.

```
nat (inside) 1 10.1.0.0 255.255.255.0
nat (inside) 2 10.1.1.0 255.255.255.0
global (outside) 1 192.168.1.1 netmask 255.255.255.0
global (outside) 2 209.165.200.225 netmask 255.255.255.224
```

The following example configures two port addresses for setting up PAT on hosts from the internal network 10.1.0.0/16 in global configuration mode.

```
nat (inside) 1 10.1.0.0 255.255.0.0
global (outside) 1 209.165.200.225 netmask 255.255.255.224
global (outside) 1 192.168.1.1 netmask 255.255.255.0
```

With this configuration, address 192.168.1.1 will only be used when the port pool from address 209.165.200.225 is at maximum capacity.

PAT and DNS

IP addresses in the pool of global addresses specified with the **global** command require reverse DNS entries to ensure that all external network addresses are accessible through the PIX Firewall. To create reverse DNS mappings, use a DNS PTR record in the address-to-name mapping file for each global address. For more information on DNS, refer to *DNS and BIND*, by Paul Albitz and Cricket Liu, O'Reilly & Associates, Inc., ISBN 1-56592-010-4. Without the PTR entries, sites can experience slow or intermittent Internet connectivity and FTP requests that consistently fail. For example, if a global IP address is 209.165.201.1 and the domain for the PIX Firewall is pix.example.com, the PTR record would be as follows.

```
1.201.165.209.in-addr.arpa. IN PTR pix.example.com
```

A DNS server on a higher level security interface needing to get updates from a root name server on the outside interface cannot use PAT. Instead, a **static** command statement must be added to map the DNS server to a global address on the outside interface.

For example, PAT is enabled with these commands:

```
nat (inside) 1 192.168.1.0 255.255.255.0
global (inside) 1 209.165.202.128 netmask 255.255.255.224
```

However, a DNS server on the inside at IP address 192.168.1.5 cannot correctly reach the root name server on the outside at IP address 209.165.202.130.

To ensure that the inside DNS server can access the root name server, insert the following **static** command statement:

```
static (inside,outside) 209.165.202.129 192.168.1.5
```

The global address 209.165.202.129 provides a translated address for the inside server at IP address 192.168.1.5.

Examples

The following example declares two global pool ranges and a PAT address. Then the **nat** command permits all inside users to start connections to the outside network:

```
global (outside) 1 209.165.201.1-209.165.201.10 netmask 255.255.255.224
global (outside) 1 209.165.201.12 netmask 255.255.255.224
Global 209.165.201.12 will be Port Address Translated
nat (inside) 1 0 0
clear xlate
```

The next example creates a global pool from two contiguous pieces of a Class C address and gives the perimeter hosts access to this pool of addresses to start connections on the outside interface:

```
global (outside) 1000 209.165.201.1-209.165.201.14 netmask 255.255.255.240
global (outside) 1000 209.165.201.17-209.165.201.30 netmask 255.255.255.240
nat (perimeter) 1000 0 0
```

help

Display help information.

help *command*

?

Syntax Description

?	Displays all commands available in the current privilege level and mode.
command	Specifies the PIX Firewall command for which to display the PIX Firewall command-line interface (CLI) help.
help	If no command name is specified, displays all commands available in the current privilege level and mode; otherwise, displays the PIX Firewall CLI help for the command specified.

Command Modes

Unprivileged, privileged, and configuration modes.

Usage Guidelines

The **help** or **?** command displays help information about all commands. You can view help for an individual command by entering the command name followed by a "**?**"(question mark).

If the **pager** command is enabled and when 24 lines display, the listing pauses, and the following prompt appears:

```
<--- More --->
```

The More prompt uses syntax similar to the UNIX **more** command:

- To view another screenful, press the Space bar.
- To view the next line, press the **Enter** key.
- To return to the command line, press the **q** key.

Examples

The following example shows how you can display help information by following the command name with a question mark:

```
enable ?
usage: enable password <pw> [encrypted]
```

Help information is available on the core commands (not the **show**, **no**, or **clear** commands) by entering **?** at the command prompt:

```
?
aaa Enable, disable, or view TACACS+ or RADIUS
 user authentication, authorization and accounting
…
```

hostname

Change the host name in the PIX Firewall command-line prompt.

hostname *newname*

Syntax Description

newname	Specifies a new host name for the firewall and is displayed in the firewall prompt. This name can be up to 63 characters, including alphanumeric characters, spaces or any of the following special characters: ‘( ) + - , . / : = ?

Command Modes

Configuration mode.

Usage Guidelines

The **hostname** command changes the host name label on prompts. The default host name is pixfirewall.

Note The change of the host name causes the change of the fully qualified domain name. Once the fully qualified domain name is changed, delete the RSA key pairs with the **ca zeroize rsa** command and delete related certificates with the **no ca identity** *ca_nickname* command.

Examples

The following example shows how to change a host name:

```
pixfirewall(config)# hostname spinner
spinner(config)# hostname pixfirewall
pixfirewall(config)#
```

http

Enables the PIX Firewall HTTP server and specifies the clients that are permitted to access it. Additionally, for access, the Cisco PIX Device Manager (PDM) requires that the PIX Firewall have an enabled HTTP server.

[**no**] **http** *ip_address* [*netmask*] [*if_name*]

[**no**] **http server enable**

clear http

show http

Syntax Description

clear http	Removes all HTTP hosts and disables the server.
http	Relating to the Hypertext Transfer Protocol.
http server enable	Enables the HTTP server required to run PDM.
if_name	PIX Firewall interface name on which the host or network initiating the HTTP connection resides.
ip_address	Specifies the host or network authorized to initiate an HTTP connection to the PIX Firewall.
netmask	Specifies the network mask for the **http** *ip_address*.

Defaults

If you do not specify a netmask, the default is **255.255.255.255** regardless of the class of IP address. The default *if_name* is **inside**.

Command Modes

Configuration mode.

Usage Guidelines

Access from any host will be allowed if **0.0.0.0 0.0.0.0** (or **0 0**) is specified for *ip_address* and *netmask*.

The **show http** command displays the allowed hosts and whether or not the HTTP server is enabled.

Examples

The following **http** command example is used for one host:

```
http 16.152.1.11 255.255.255.255 outside
```

The following **http** command example is used for any host:

```
http 0.0.0.0 0.0.0.0 inside
```

icmp

Configure access rules for Internet Control Message Protocol (ICMP) traffic that terminates at an interface.

[no] icmp {**permit** | **deny**} *ip_address net_mask* [*icmp_type*] *if_name*

clear icmp

show icmp

Syntax Description

deny	Deny access if the conditions are matched.
icmp_type	ICMP message type as described in Table 6-1.
if_name	The interface name.
ip_address	The IP address of the host sending ICMP messages to the interface.
net_mask	The mask to be applied to *ip_address*.
permit	Permit access if the conditions are matched.

Command Modes

Configuration mode.

Usage Guidelines

By default, the PIX Firewall denies all inbound traffic through the outside interface. Based on your network security policy, you should consider configuring the PIX Firewall to deny all ICMP traffic at the outside interface, or any other interface you deem necessary, by using the **icmp** command.

The **icmp** command controls ICMP traffic that received by the firewall. If no ICMP control list is configured, then the PIX Firewall accepts all ICMP traffic that terminates at any interface (including the outside interface), except that the PIX Firewall does not respond to ICMP echo requests directed to a broadcast address.

The **icmp deny** command disables pinging to an interface, and the **icmp permit** command enables pinging to an interface. With pinging disabled, the PIX Firewall cannot be detected on the network. This is also referred to as configurable proxy pinging.

For traffic that is routed through the PIX Firewall only, you can use the **access-list** or **access-group** commands to control the ICMP traffic routed through the PIX Firewall.

We recommend that you grant permission for ICMP unreachable message type (type 3). Denying ICMP unreachable messages disables ICMP Path MTU discovery, which can halt IPSec and PPTP traffic. See RFC 1195 and RFC 1435 for details about Path MTU Discovery.

If an ICMP control list is configured, then the PIX Firewall uses a first match to the ICMP traffic followed by an implicit deny all. That is, if the first matched entry is a permit entry, the ICMP packet continues to be processed. If the first matched entry is a deny entry or an entry is not matched, PIX Firewall discards the ICMP packet and generates the %PIX-3-313001 syslog message. An exception is when an ICMP control list is not configured; in that case, a permit is assumed.

The syslog message is as follows:

```
%PIX-3-313001: Denied ICMP type=type, code=code from source_address on interface
interface_number
```

If this message appears, contact the peer's administrator.

ICMP Message Types

Table 6-1 lists possible ICMP type values.

Table 6-1 ICMP Type Literals

ICMP Type	Literal
0	echo-reply
3	unreachable
4	source-quench
5	redirect
6	alternate-address
8	echo
9	router-advertisement
10	router-solicitation
11	time-exceeded
12	parameter-problem
13	timestamp-request
14	timestamp-reply
15	information-request
16	information-reply
17	mask-request
18	mask-reply
31	conversion-error
32	mobile-redirect

Examples

1. Deny all ping requests and permit all unreachable messages at the outside interface:

```
icmp permit any unreachable outside
```

The default behavior of the PIX Firewall is to deny ICMP messages to the outside interface.

2. Permit host 172.16.2.15 or hosts on subnet 172.22.1.0/16 to ping the outside interface:

```
icmp permit host 172.16.2.15 echo-reply outside
icmp permit 172.22.1.0 255.255.0.0 echo-reply outside
icmp permit any unreachable outside
```

igmp

Refer to the **multicast** command for the **igmp** subcommands.

The Internet Group Management Protocol (IGMP) enables IP hosts to report their multicast group memberships to an adjacent multicast router. On the PIX Firewall, IGMP support is implemented as a subcommand to the **multicast** command.

interface

Sets network interface parameters and configures VLANs.

interface *hardware_id* [*hardware_speed* [**shutdown**]]

[**no**] **interface** *hardware_id vlan_id* [**logical | physical**] [**shutdown**]

interface *hardware_id* **change-vlan** *old_vlan_id new_vlan_id*

clear interface

show interface *hardware_id* [*hardware_speed*] [**shutdown**]

Syntax Description

change-vlan	Keyword to change the VLAN identifier for an interface.
hardware_id	Identifies the network interface type. Possible values are **ethernet0**, **ethernet1** to **ethernet***n*, or **gb-ethernet***n,* depending on how many network interfaces are in the PIX Firewall.
hardware_speed	Network interface speed (optional). **aui**—Set 10 for Mbps Ethernet half-duplex communication with an AUI cable interface. **auto**—Negotiates Ethernet speed and duplex settings automatically. The **auto** keyword can only be used with the Intel 10/100 automatic speed-sensing network interface card. **bnc**—Set for 10 Mbps Ethernet half-duplex communication with a BNC cable interface. Possible Ethernet values are: **10baseT**—To set for 10 Mbps Ethernet half-duplex communication. **10full**—To set for 10 Mbps Ethernet full-duplex communication. **100baseTX**—To set for 100 Mbps Ethernet half-duplex communication. **100full**—To set for 100 Mbps Ethernet full-duplex communication. Possible Gigabit Ethernet (gb-ethernetX) values are: **1000auto**—To auto negotiate speed and duplex. **1000full**—To auto negotiate, advertising 1000 Mbps full duplex. **1000full nonegotiate**—To force link to 1000 Mbps full duplex.
logical	Creates a logical interface and applies the VLAN.
new_vlan_id	The new VLAN indentifier.
old_vlan_id	The current VLAN indentifier.
physical	Apply VLAN to physical interface.
shutdown	Disable an interface.
vlan_id	The VLAN identifier. For example: vlan10, vlan20, and so on.

Command Modes

Configuration mode.

Defaults

When configured, VLAN logical interfaces are enabled by default.

Usage Guidelines

The **interface** command sets the speed and duplex settings of the network interface boards, and brings up the interfaces specified. After changing an **interface** command, use the **clear xlate** command.

Note For Stateful Failover to work properly, set the Stateful Failover dedicated interface to 100 Mbps full duplex using the **100full** option to the **interface** command.

The i82542 Gigabit Ethernet interface currently used in the PIX Firewall does not support half duplex; as a result, **1000auto** is equivalent to **1000full** when using this interface.

VLAN interfaces

With Version 6.3, you can assign VLANs to physical interfaces on the PIX Firewall, or you can configure multiple logical interfaces on a single physical interface and assign each logical interface to a specific VLAN.

Physical interfaces are one per each NIC, in place at boot time, and non-removable. Logical interfaces that can be many-to-one for each NIC, are created at run time, and can be removed through software reconfiguration. A minimum of two physical interfaces are required for all PIX Firewall platforms to support VLANs.

A logical interface is similar in many respects to a so-called physical interface. Both logical and physical interfaces are software objects (the actual *physical* object is the network interface card on the PIX Firewall unit). What is called the physical interface for the purpose of configuration is a software object that has both Layer 2 (Data link) and Layer 3 (Network) attributes. Layer 2 attributes include maximum transmission unit (MTU) size and failover status, while Layer 3 attributes include IP address and security level.

A logical interface has only Layer 3 attributes. As a result, you can issue certain commands, such as **failover link** *if_name* or **failover lan interface** *if_name* on a physical interface that you cannot use with a logical interface. When you disable a physical interface, all the associated logical interfaces are also disabled. When you disable a logical interface, it only affects the logical interface.

The number of logical interfaces that you can configure varies according to the model. The minimum number of interfaces for any PIX Firewall is two. Table 6-2 lists the maximum number of logical interfaces supported on a specific PIX Firewall model:

Table 6-2 Maximum Number of Interfaces Supported on PIX Firewall Models

Model	Restricted License[1]			Unrestricted License		
	Total Interfaces	Physical Interfaces	Logical Interfaces	Total Interfaces	Physical Interfaces	Logical Interfaces
PIX 501[2]	NA	NA	NA	2	2	Not supported
PIX 506/506E	NA	NA	NA	2	2	Not supported
PIX 515/515E	5	3	3	10	6	8
PIX 520[3]	NA	NA	NA	12	6	10

Table 6-2 Maximum Number of Interfaces Supported on PIX Firewall Models (continued)

Model	Restricted License[1]			Unrestricted License		
	Total Interfaces	Physical Interfaces	Logical Interfaces	Total Interfaces	Physical Interfaces	Logical Interfaces
PIX 525	8	6	6	12	8	10
PIX 535	10	8	8	24	10	22

1. PIX 501 and PIX 506/506E do not support Restricted/Unrestricted licenses.
2. One interface of the PIX 501 connects to an integrated 4-port switch.
3. PIX 520 supports a connection license and the number of interfaces does not vary with the connection license.

Note To determine the maximum number of logical interfaces that you can use, subtract the number of physical interfaces in use on your PIX Firewall from the number of total interfaces.

Use the **show interface** command to display information about the VLAN configuration.

Use the **interface** *hardware_id vlan_id* **logical shutdown** command to temporarily disable a logical interface.

Use the **interface** *hardware_id* **change-vlan** *old_vlan_id new_vlan_id* command to reassign a VLAN.

Use the **no interface** *hardware_id vlan_id* **logical** command to remove the VLAN configuration.

no and clear commands

The **clear interface** command clears all interface statistics except the number of input bytes. This command no longer shuts down all system interfaces. The **clear interface** command works with all interface types except Gigabit Ethernet. The **clear interface** command also clears the packet drop count of Unicast RPF for all interfaces.

Use the **no interface** command to remove logical interfaces and VLAN definitions. (However, a **no interface** command does not negate an interface **shutdown** command.)

Note Using a **no interface** command on a logical interface (used for VLAN configuration) removes the logical interface from the system. Removing the logical interface also deletes all configuration rules applied to that interface, so exercise caution when using **no interface** commands with logical interfaces.

The **shutdown** option lets you disable an interface. When you first install PIX Firewall, all interfaces are shut down by default. You must explicitly enable an interface by entering the command without the **shutdown** option. If the **shutdown** option does not exist in the command, packets are passed by the driver to and from the card.

If the **shutdown** option does exist, packets are dropped in either direction. Inserting a new card defaults to the default interface command containing the **shutdown** option. (That is, if you add a new card and then enter the **write memory** command, the **shutdown** option is saved into Flash memory for the interface.) When upgrading from a previous version to the current version, interfaces are enabled.

The configuration of the interface affects buffer allocation (the PIX Firewall will allocate more buffers for higher line speeds). Buffer allocation can be checked with the **show blocks** command.

Note Even though the default is to set automatic speed sensing for the interfaces with the **interface** *hardware_id* **auto** command, we recommend that you specify the speed of the network interfaces; for example, **10baseT** or **100baseTX**. This lets PIX Firewall operate in network environments that may include switches or other devices that do not handle auto sensing correctly.

show interface

The **show interface** command lets you view network interface information for Ethernet. This is one of the first commands you should use when establishing network connectivity after installing a PIX Firewall.

Note The PIX 501 switch interface always indicates `100000 Kbit full duplex` (100,000 Kbps full duplex) even though the switch ports have negotiated the speed and duplex settings. The PIX Firewall automatically negotiates the inside interface setting at **100full** and this is not configurable.

Gigabit interface cards do not provide information for the extended **show interface** command counters introduced in Version 5.0(3). For Gigabit Ethernet interfaces, the current and maximum count for the number of blocks on the input (receive) queue will always be the same (63).

The information in the **show interface** command is as follows in Table 6-3:

Table 6-3 show interface Description

Show Interface Command Output	Description
Ethernet string	Indicates that you have used the **interface** command to configure the interface. The statement indicates either outside or inside, and whether the interface is available (“up”) or not available (“down”).
line protocol up or line protocol down	The message “line protocol up” means a working cable is plugged into the network interface. If the message is “line protocol down,” either the cable is incorrect or not plugged into the interface connector. The **show interface** command reports “line protocol down” for BNC cable connections and for 3Com cards.
Network interface type	Indicates type of network interface.
Interrupt vector	Note: It is acceptable for interface cards to have the same interrupts.
MAC address	Intel cards start with “i” and 3Com cards with “3c.”
Maximum transmission unit (MTU)	The size, in bytes, that data can best be sent over the network.
nn packets input	Indicates that packets are being received in the PIX Firewall.
nn packets output	Indicates that packets are being sent from the PIX Firewall.
Line duplex status	Half duplex indicates that the network interface switches back and forth between sending and receiving information; full duplex indicates that the network interface can send or receive information simultaneously.
Line speed	**10baseT** is listed as 10,000 Kb; **100baseTX** is listed as 100,000 Kb.

- The **show interface** command includes eight status counters (valid only for Ethernet interfaces).

  The following example shows sample output:

```
show interface
interface ethernet0 "outside" is up, line protocol is up
  Hardware is i82559 ethernet, address is 00aa.0000.003b
  IP address 209.165.201.7, subnet mask 255.255.255.224
  MTU 1500 bytes, BW 100000 Kbit half duplex
        1184342 packets input, 1222298001 bytes, 0 no buffer
        Received 26 broadcasts, 27 runts, 0 giants
        4 input errors, 0 CRC, 4 frame, 0 overrun, 0 ignored, 0 abort
        1310091 packets output, 547097270 bytes, 0 Andorrans, 0 unicast repave drops
        0 output errors, 28075 collisions, 0 interface resets
        0 babbles, 0 late collisions, 117573 deferred
        0 lost carrier, 0 no carrier
input queue (cure/max blocks): hardware (128/128) software (0/1)
        output queue (cure/max blocks): hardware (0/2) software (0/1)
```

The **show interface** counter descriptions are as follows in Table 6-4:

Table 6-4 show interface Counters

Counter	Description
output errors	The number of frames not transmitted because the configured maximum number of collisions was exceeded. This counter should only increment during heavy network traffic.
collisions	The number of messages retransmitted due to an Ethernet collision (single and multiple collisions). This usually occurs on an overextended LAN (Ethernet or transceiver cable too long, more than two repeaters between stations, or too many cascaded multiport transceivers). A packet that collides is counted only once by the output packets.
interface resets	The number of times an interface has been reset. If an interface is unable to transmit for three seconds, PIX Firewall resets the interface to restart transmission. During this interval, connection state is maintained. An interface reset can also happen when an interface is looped back or shut down.
babbles	Unused. (“babble” means that the transmitter has been on the interface longer than the time taken to transmit the largest frame.)
late collisions	The number of frames that were not transmitted because a collision occurred outside the normal collision window. A late collision is a collision that is detected late in the transmission of the packet. Normally, these should never happen. When two Ethernet hosts try to talk at once, they should collide early in the packet and both back off, or the second host should see that the first one is talking and wait. If you get a late collision, a device is jumping in and trying to send the packet on the Ethernet while the PIX Firewall is partly finished sending the packet. The PIX Firewall does not resend the packet, because it may have freed the buffers that held the first part of the packet. This is not a real problem because networking protocols are designed to cope with collisions by resending packets. However, late collisions indicate a problem exists in your network. Common problems are large repeated networks and Ethernet networks running beyond the specification.

Table 6-4 show interface Counters (continued)

Counter	Description
deferred	The number of frames that were deferred before transmission due to activity on the link.
lost carrier	The number of times the carrier signal was lost during transmission.
no carrier	Unused.
input queue (curr/max blocks)	Input queue—The input (receive) hardware and software queue. • hardware—(current and maximum blocks). The number of blocks currently present on the input hardware queue, and the maximum number of blocks previously present on that queue. In the example, there are currently 128 blocks on the input hardware queue, and the maximum number of blocks ever present on this queue was 128. • software—(current and maximum blocks). The number of blocks currently present on the input software queue, and the maximum number of blocks previously present on that queue. In the example, there are currently 0 blocks on the input software queue, and the maximum number of blocks ever present on this queue was 1.
output queue (curr/max blocks)	Output queue—The output (transmit) hardware and software queue. • hardware—(current and maximum blocks). The number of blocks currently present on the output hardware queue, and the maximum number of blocks previously present on that queue. In the example, there are currently 0 blocks on the output hardware queue, and the maximum number of blocks ever present on this queue was 2. • software—(current and maximum blocks). The number of blocks currently present on the output software queue, and the maximum number of blocks previously present on that queue. In the example, there are currently 0 blocks on the output software queue, and the maximum number of blocks ever present on this queue was 1.

Examples

The following example shows interface activity on the interface ethernet0, which has been named **outside**:

```
show interface
interface ethernet0 "outside" is up, line protocol is up
  Hardware is i82559 ethernet, address is 0000.0001.0001
  IP address 209.165.201.17, subnet mask 255.255.255.0
  MTU 1500 bytes, BW 10000 Kbit full duplex
        4203 packets input, 376390 bytes, 0 no buffer
        Received 3894 broadcasts, 0 runts, 0 giants
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
        1320 packets output, 123652 bytes, 0 underruns
        0 output errors, 0 collisions, 0 interface resets
        0 babbles, 0 late collisions, 0 deferred
        0 lost carrier, 0 no carrier
        input queue (curr/max blocks): hardware (35/128) software (0/2)
        output queue (curr/max blocks): hardware (0/1) software (0/1)
```

The following example sets a Gigabit Ethernet interface named **gb0** to **1000full nonegotiate**:

```
pixfirewall(config)# interface gb0 1000full nonegotiate
```

Sample output from the subsequent **show interface** command is as follows:

```
pixfirewall(config)# show interface gb0
interface gb-ethernet0 "intf2" is up, line protocol is down
Hardware is i82543 rev02 gigabit ethernet, address is 0003.47df.1e1c
MTU 1500 bytes, BW 1 Gbit full duplex, Force link-up
        5133 packets input, 628176 bytes, 0 no buffer
        Received 4202 broadcasts, 2 runts, 8 giants
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
        1832 packets output, 124948 bytes, 0 underruns
        input queue (curr/max blocks): hardware (41/128) software (0/2)
        output queue (curr/max blocks): hardware (0/2) software (0/4)
```

The "`Force link-up`" keyword indicates that the link was forced and not negotiated.

The following is sample output from the **show interface** command on a PIX 501. Notice that the interface speed and settings are always displayed as `100000 Kbit half duplex`.

```
pixfirewall(config)# show interface
interface ethernet0 "outside" is up, line protocol is up
  Hardware is i82559 ethernet, address is 0007.eb9b.56aa
  MTU 1500 bytes, BW 100000 Kbit half duplex
        114 packets input, 6840 bytes, 0 no buffer
        Received 114 broadcasts, 0 runts, 0 giants
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
        62982 packets output, 78915110 bytes, 0 underruns
        0 output errors, 0 collisions, 0 interface resets
        0 babbles, 0 late collisions, 0 deferred
        1483 lost carrier, 0 no carrier
        input queue (curr/max blocks): hardware (128/128) software (0/1)
        output queue (curr/max blocks): hardware (0/115) software (0/64)
interface ethernet1 "inside" is up, line protocol is up
  Hardware is i82559 ethernet, address is 0007.eb9b.56ab
  IP address 192.168.1.1, subnet mask 255.255.255.0
  MTU 1500 bytes, BW 100000 Kbit full duplex
        55005197 packets input, 903916376 bytes, 0 no buffer
        Received 0 broadcasts, 0 runts, 0 giants
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
        2 packets output, 120 bytes, 0 underruns
        0 output errors, 0 collisions, 0 interface resets
        0 babbles, 0 late collisions, 0 deferred
        0 lost carrier, 0 no carrier
        input queue (curr/max blocks): hardware (128/128) software (0/59)
        output queue (curr/max blocks): hardware (0/1) software (0/1)
```

Related Commands

nameif	Assigns a name to an interface.
ip address	Configures the IP address and mask for an interface, or defines a local address pool.

ip address

Identifies addresses for network interfaces, and enables you to set the number of times the PIX Firewall will poll for DHCP information.

[no] ip address *if_name ip_address* [*netmask*]

[no] ip address outside dhcp [**setroute**] [**retry** *retry_cnt*]

[**no**] **ip address** *if_name* **pppoe** [**setroute**]

[**no**] **ip address** *if_name ip_address netmask* **pppoe** [**setroute**]

clear ip

show ip

show ip address *if_name* **dhcp**

show ip address *if_name* **pppoe**

Syntax Description

clear ip	Clears all interface IP addresses. The **clear ip** command does not affect the **ip local pool** or **ip verify reverse-route** commands.
dhcp	Specifies PIX Firewall will use DHCP to poll for information. Enables the DHCP client feature on the specified interface.
if_name	The internal or external interface name designated by the **nameif** command.
ip_address	PIX Firewall unit's network interface IP address. Each interface IP address must be unique. Two or more interfaces must not be given the same IP address or IP addresses which are on the same IP network.
netmask	Network mask of *ip_address*.
outside	Interface from which the PIX Firewall will poll for information.
pppoe	Specifies to use Point-to-Point Protocol over Ethernet (PPPoE) to assign an IP address.
retry	Enables PIX Firewall to retry a poll for DHCP information.
retry_cnt	Specifies the number of times PIX Firewall will poll for DHCP information. The values available are 4 to 16. If no value is specified, the default is 4.
setroute	This option tells the PIX Firewall to set the default route using the default gateway parameter the DHCP or PPPoE server returns.

Command Modes

Configuration mode.

Defaults

By default, the PIX Firewall will not retry to poll for DHCP information. The default value for *retry_cnt* is **4**.

Usage Guidelines

The **ip address** command lets you assign an IP address to each interface.

Note Each interface IP address must be unique and not on the same network as any another interface on the firewall.

Use the **show ip** command to view which addresses are assigned to the network interfaces. If you make a mistake while entering this command, reenter the command with the correct information. The **clear ip** command clears all interface IP addresses. The **clear ip** command does not affect the **ip local pool** or **ip verify reverse-route** commands.

Note The **clear ip** command stops all traffic through the PIX Firewall unit.

After changing an **ip address** command, use the **clear xlate** command.

Always specify a network mask with the **ip address** command. If you let PIX Firewall assign a network mask based on the IP address, you may not be permitted to enter subsequent IP addresses if another interface's address is in the same range as the first address. For example, if you specify an inside interface address of 10.1.1.1 without specifying a network mask and then try to specify 10.1.2.2 for a perimeter interface mask, PIX Firewall displays the error message, "Sorry, not allowed to enter IP address on same network as interface *n*." To fix this problem, reenter the first command specifying the correct network mask.

Do not set the netmask to all 255s, such as 255.255.255.255. This stops access on the interface. Instead, use a network address of 255.255.255.0 for Class C addresses, 255.255.0.0 for Class B addresses, or 255.0.0.0 for Class A addresses.

PIX Firewall configurations using failover require a separate IP address for each network interface on the standby unit. The system IP address is the address of the active unit. When the **show ip** command is executed on the active unit, the current IP address is the same as the system IP address. When the **show ip** command is executed on the standby unit, the system IP address is the failover IP address configured for the standby unit.

Note If an IP address has not been configured for a physical or VLAN interface, or the IP address for the interface has been deleted using the **clear ip** command, the IP address for that interface is no longer set to 127.0.0.1 by default. In this case, the interface is does not have an IP address.

Note When using the IP address of an interface as the device ID in logging messages sent to a syslog server and the IP address of that interface is cleared, the device ID uses 0.0.0.0.

show ip address commands

The **show ip** command isplays IP addresses assigned to the network interfaces.

The **show ip address** *if_name* **dhcp** command displays detailed information about the DHCP lease.

The **show ip address** *if_name* **pppoe** command displays detailed information about the PPPOE connection.

DHCP client

The **ip address dhcp** command enables the DHCP client feature within the PIX Firewall. This command allows the PIX Firewall to be a DHCP client to a DHCP server that provides configuration parameters to the client. In this case, the configuration parameters the DHCP server provides is an IP address and a subnet mask to the interface on which the DHCP client feature is enabled. The optional **setroute** argument tells the PIX Firewall to set the default route using the default gateway parameter the DHCP server returns. If the **setroute** argument is configured, the **show route** command output shows the default route as being set by a DHCP server. To reset the interface and delete the DHCP lease from PIX Firewall, configure a static IP address with the **ip address** *if_name ip_address* [*netmask*] or **ip address** *if_name* **pppoe | dhcp** [**setroute**] command, or use the **clear ip** command.

The **ip address dhcp** and **pppoe** command options are mutually exclusive.

Note Do not configure the PIX Firewall with a default route when using the **setroute** argument of the **ip address dhcp** or **ip address pppoe** command.

PPPoE client

The PPPoE client functionality is turned off by default, and you must first use the **vpdn** commands to configure the PIX Firewall for PPPoE; the **vpdn** commands set the username, password, and authentication protocol for PPPoE access.

PPPoE is only supported on the PIX Firewall outside interface in PIX Firewall software Version 6.2.

The **ip address pppoe** command enables the PPPoE client feature within the PIX Firewall. (You can also use this command to clear and restart a PPPoE session; the current session shuts down and a new one restarts after entering this command.) You must enter the PPPoE configuration using the **vpdn** commands before enabling PPPoE with the **ip address pppoe** command.

You can also enable PPPoE by manually entering the IP address, using the **ip address** *if_name ip_address netmask* **pppoe** command. This command sets the PIX Firewall to use the specified address instead of negotiating with the PPPoE server to assign an address.

The **ip address setroute** command enables an access concentrator to set the default routes for the PPPoE client.

The **ip address pppoe** and **dhcp** command options are mutually exclusive.

For more information

See the *Cisco PIX Firewall and VPN Configuration Guide* for more information about the DHCP and PPPoE client features.

Examples

The following is sample output from the **show ip** command:

```
show ip
System IP Addresses:
        ip address outside 209.165.201.2 255.255.255.224
        ip address inside 192.168.2.1 255.255.255.0
        ip address perimeter 192.168.70.3 255.255.255.0
Current IP Addresses:
        ip address outside 209.165.201.2 255.255.255.224
        ip address inside 192.168.2.1 255.255.255.0
        ip address perimeter 192.168.70.3 255.255.255.0
```

The Current IP Addresses are the same as the System IP Addresses on the failover active unit. When the primary unit fails, the Current IP Addresses become those of the standby unit.

The following is sample output from the **show ip address dhcp** command:

```
show ip address outside dhcp
Temp IP Addr:209.165.201.57 for peer on interface:outside
Temp sub net mask:255.255.255.224
DHCP Lease server:209.165.200.225, state:3 Bound
DHCP Transaction id:0x4123
Lease:259200 secs, Renewal:129600 secs, Rebind:226800 secs
Temp default-gateway addr:209.165.201.1
Next timer fires after:111797 secs
Retry count:0, Client-ID:cisco-0000.0000.0000-outside

ip address outside dhcp retry 10
```

Related Commands

dhcpd	Configures the DHCP server.
vpdn	Configures VPDN (PPTP, L2TP, PPPoE) policy.

ip audit

Configures IDS signature use.

[**no**] **ip audit attack** [**action** [**alarm**] [**drop**] [**reset**]]

[**no**] **ip audit info** [action [alarm] [drop] [reset]]

[**no**] **ip audit interface** *if_name audit_name*

[**no**] **ip audit name** *audit_name* **attack** [**action** [**alarm**] [**drop**] [**reset**]]

[**no**] **ip audit name** *audit_name* **info** [**action** [**alarm**] [**drop**] [**reset**]]

[**no**] **ip audit signature** *signature_number* **disable**

show ip audit count [**global**] [**interface** *interface*]

show ip audit {**info** | **attack**}

show ip audit interface [*if_name*]

show ip audit name [*audit_name* [**info**|**attack**]]

show ip audit signature [*signature_number*]

clear ip audit [**configuration**]

clear ip audit count [**global** | **interface** *interface*]

Syntax Description

action [**alarm**] [**drop**] [**reset**]	The **alarm** option reports to all configured syslog servers that a signature match is detected in a packet. The **drop** option drops the offending packet. The **reset** option drops the offending packet and closes the connection if it is part of an active connection. The default is **alarm**. When no option is specified (you enter "`ip audit info action`" only), all actions are disabled.
audit attack	Specify the default actions to be taken for attack signatures.
audit info	Specify the default actions to be taken for informational signatures or disable all actions.
audit interface	Apply an audit specification or policy (via the **ip audit name** command) to an interface.
audit name	Specify informational signatures, except those disabled or excluded by the **ip audit signature** command, as part of the policy.
audit signature	Specify which messages to display, attach a global policy to a signature, and disable or exclude a signature from auditing.
audit_name	Audit policy name viewed with the **show ip audit name** command.
clear	Resets **name, signature, interface, attack, info** to their default values**.**

configuration	The already configured **ip audit** commands.
count	The number of signature matches.
global	All firewall interfaces.
interface *interface*	The name of a firewall interface, defined by the **nameif** command.
signature_number	An IDS signature number.

Command Modes

Configuration mode.

Usage Guidelines

Cisco Intrusion Detection System (Cisco IDS) provides the following for IP-based systems:

- Traffic auditing. Application-level signatures will only be audited as part of an active session.
- Applies the audit to an interface.
- Supports different audit policies. Traffic matching a signature triggers a range of configurable actions.
- Disables the signature audit.
- Enables IDS and still disables actions of a signature class (informational, attack).

Auditing is performed by looking at the IP packets as they arrive at an input interface, if a packet triggers a signature and the configured action does not drop the packet, then the same packet can trigger other signatures.

PIX Firewall supports both inbound and outbound auditing.

For a complete list of supported Cisco IDS signatures, their wording, and whether they are attack or informational messages, refer to *Cisco PIX Firewall System Log Messages*.

Refer to the *Cisco Secure Intrusion Detection System Version 2.2.1 User Guide* for detailed information on each signature. You can view the "NSDB and Signatures" chapter of this guide at the following website:

http://www.cisco.com/univercd/cc/td/doc/product/iaabu/csids/csids1/csidsug/sigs.htm

The **ip audit** commands are described in the sections that follow.

ip audit attack

The **ip audit attack** [**action** [**alarm**] [**drop**] [**reset**]] command specifies the default actions to be taken for attack signatures. An audit policy (audit rule) defines the attributes for all signatures that can be applied to an interface along with a set of actions. Using an audit policy may limit the traffic that is audited or specify actions to be taken when the signature matches. Each audit policy is identified by a name and can be defined for informational or attack signatures. Each interface can have two policies; one for informational signatures and one for attack signatures. If a policy is defined without actions, then the configured default actions will take effect. Each policy requires a different name.

The **no ip audit attack** command resets the action to be taken for attack signatures to the default action.

ip audit info

The **ip audit info** [**action** [**alarm**] [**drop**] [**reset**]] command specifies the default action to be taken for signatures classified as informational signatures. The **ip audit info action** command disables all actions. For example,

```
pixfirewall(config)# ip audit info action
Warning: no actions specified. All actions disabled.
```

The **no ip audit info** command sets the action to be taken for signatures classified as informational and reconnaissance to the default action.

ip audit interface

The **ip audit interface** *if_name audit_name* command applies an audit specification or policy (via the ip audit name command) to an interface. The **no ip audit interface** [*if_name*] command removes a policy from an interface.

ip audit name

The **ip audit name** *audit_name* **info** [**action** [**alarm**] [**drop**] [**reset**]] command specifies the informational signatures except those disabled or excluded by the **ip audit signature** command that are considered part of the policy. The **no ip audit name** *audit_name* [**info**] command removes the audit policy *audit_name*.

ip audit signature

The **ip audit signature** *signature_number* **disable** command specifies which messages to display, attaches a global policy to a signature, and disables or excludes a signature from auditing. The **no ip audit signature** *signature_number* command removes the policy from a signature. It is used to reenable a signature.

show ip audit commands

The **show ip audit attack** command displays the default attack actions.

The **show ip audit info** command displays the default informational actions.

The **show ip audit interface** command displays the interface configuration.

The **show ip audit name** command displays all audit policies or specific policies referenced by name and type where possible.

The **show ip audit signature** command displays disabled signatures.

Supported IDS Signatures

PIX Firewall lists the following single-packet IDS signature messages: 1000-1006, 1100, 1102, 1103, 2000-2012, 2150, 2151, 2154, 3040-3042, 4050-4052, 6050-6053, 6100-6103, 6150-6155, 6175, 6180, and 6190. All signature messages are not supported by PIX Firewall in this release. IDS syslog messages all start wit

PIX-4-4000*nn* and have the following format:

```
%PIX-4-4000nn IDS:sig_num sig_msg from faddr to laddr on interface int_name
```

where the options are as follows:

sig_num	The signature number.
sig_msg	The signature message—approximately the same as the Cisco IDS signature message.
faddr	The IP address of the foreign host initiating the attack. (“Foreign” is relative; attacks can be perpetrated either from outside to an inside host, or from the inside to an outside host.)
laddr	The IP address of the local host to which the attack is directed. (“Local” is relative; attacks can be perpetrated either from the outside to an inside host, or from the inside to an outside host.)
int_name	The name of the interface on which the signature originated.

For example:

```
%PIX-4-400013 IDS:2003 ICMP redirect from 10.4.1.2 to 10.2.1.1 on interface dmz
%PIX-4-400032 IDS:4051 UDP Snork attack from 10.1.1.1 to 192.168.1.1 on interface outside
```

Examples

The following example disables the signature 6102 globally:

```
ip audit signature 6102 disable
```

The following example specifies default informational actions:

```
ip audit name attack1 info
```

The following example specifies an attack policy:

```
ip audit name attack2 attack action alarm drop reset
```

The following example applies a policy to an interface:

```
ip audit interface outside attack1
ip audit interface inside attack2
```

ip local pool

Identify addresses for a local pool.

ip local pool *pool_name pool_start_address[-pool_end_address] [mask mask]*

clear ip local pool *pool_name ip_address*[**-***ip_address*]

show ip local pool *pool_name ip_address*[**-***ip_address*]

[**no**] **ip local pool** *pool_name pool_start-address*[**-***pool_end-address*]

clear ip local pool	Removes all **ip local pool** configurations.
ip local pool	Creates a pool of local addresses to be used for assigning dynamic IP addresses to remote VPN clients. The address range of this pool of local addresses must not overlap with any command statement that lets you specify an IP address.
ip_address	Specify as a single IP address or use with *-ip_address* to specify a range of IP addresses.
-*ip_address*	Optional ending IP address.
no ip local pool	Deletes a local address pool.
pool_name	Local pool name.
pool_start_address pool_end_address	Local pool IP address range.
[mask <mask>]	Add an optional netmask. If the netmask is configured then the PIX Firewall headend will return it to the VPN client. If the netmask is not configured, PIX Firewall will retain backward compatibility with its previous behavior by not returning the netmask. If **netmask** is not configured, the PIX Firewall will use netmask 255.255.255.0.

Command Modes Configuration mode.

Usage Guidelines The **ip local pool** command lets you create a pool of local addresses to be used for assigning dynamic IP addresses to remote VPN clients. The address range of this pool of local addresses must not overlap with any command statement that lets you specify an IP address. To delete an address pool, use the **no ip local pool** command.

When a pool of addresses set by the **ip local pool** command is empty, the following syslog message appears:

%PIX-4-404101: ISAKMP: Failed to allocate address for client from pool *poolname*

To reference this pool of local addresses, use the **isakmp client configuration address-pool** command. Refer to the *Cisco PIX Firewall and VPN Configuration Guide* for information on the **isakmp** command.

pool_name ip_address [mask <mask>]

Because newer versions of VPN Client software might not work as expected without assigning a netmask drectly through the PIX Firewall, this feature allows the user to assign the netmask to the VPN client. The operating system software will attempt to calculate what the netmask is if a netmask is not assigned, based on the class of the IP from "**ip local pool**". Using this default value may not be appropriate for network routing to work properly.

For example, 10.x.x.x. is a class A, and the windows routing table sets the netmask as 255.0.0.0.

The following example creates a pool of IP addresses and then displays the pool contents:

```
ip local pool mypool 10.0.0.10-10.0.0.20
show ip local pool mypool

Pool            Begin           End             Mask            Free    In use
mypool          10.0.0.10       10.0.0.20       Not configured    11         0

Available Addresses:
10.0.0.10
10.0.0.11
10.0.0.12
10.0.0.13
10.0.0.14
10.0.0.15
10.0.0.16
10.0.0.17
10.0.0.18
10.0.0.19
10.0.0.20
```

for **ip local pool** <*name*> <*range*> *[mask <mask>]*, a typical example range would be:

ip local pool <name> 10.1.1.1-10.1.1.254

ip verify reverse-path

Implements Unicast RPF IP spoofing protection.

ip verify reverse-path interface *int_name*

no ip verify reverse-path interface *int_name*

clear ip verify reverse-path interface *int_name*

clear ip verify

show ip verify [**reverse-path** [**interface** *int_name*]]

show ip verify statistics

Syntax Description

clear ip verify	Removes **ip verify** commands from the configuration.
clear ip verify reverse-path interface	Removes **ip verify reverse-path** commands for an individual interface from the configuration.
int_name	Name of an interface you want to protect from a DoS attack.
ip verify reverse-path interface	Protects an individual interface against IP spoofing by enabling both ingress and egress filtering to verify addressing and route integrity. This command depends upon a default route previously defined in the configuration. See RFC 2267 for more information.
no ip verify reverse-path interface	Disables **ip verify reverse-path** filtering for an individual interface from the configuration.

Command Modes

Configuration mode.

Usage Guidelines

The **ip verify reverse-path** command is a security feature that does a route lookup based on the source address. Usually, the route lookup is based on the destination address. This is why it is called reverse path forwarding. With this command enabled, packets are dropped if there is no route found for the packet or the route found does not match the interface on which the packet arrived.

The **ip verify reverse-path** command lets you specify which interfaces to protect from an IP spoofing attack using network ingress and egress filtering, which is described in RFC 2267. This command is disabled by default and provides Unicast Reverse Path Forwarding (Unicast RPF) functionality for the PIX Firewall.

The **clear ip verify** command removes **ip verify** commands from the configuration. Unicast RPF is a unidirectional input function that screens inbound packets arriving on an interface. Outbound packets are not screened.

Because of the danger of IP spoofing in the IP protocol, measures need to be taken to reduce this risk when possible. Unicast RPF, or reverse route lookup, prevents such manipulation under certain circumstances.

Note The **ip verify reverse-path** command depends on the existence of a default route statement in the configuration for the outside interface that has 0.0.0.0 0.0.0.0 in the **route** command statement for the IP address and network mask.

The **ip verify reverse-path** command provides both ingress and egress filtering. Ingress filtering checks inbound packets for IP source address integrity, and is limited to addresses for networks in the enforcing entity's local routing table. If the incoming packet does not have a source address represented by a route, then it is impossible to know whether the packet has arrived on the best possible path back to its origin. This is often the case when routing entities cannot maintain routes for every network.

Egress filtering verifies that packets destined for hosts outside the managed domain have IP source addresses verifiable by routes in the enforcing entity's local routing table. If an exiting packet does not arrive on the best return path back to the originator, then the packet is dropped and the activity is logged. Egress filtering prevents internal users from launching attacks using IP source addresses outside of the local domain because most attacks use IP spoofing to hide the identity of the attacking host. Egress filtering makes the task of tracing the origin of an attack much easier. When employed, egress filtering enforces what IP source addresses are obtained from a valid pool of network addresses. Addresses are kept local to the enforcing entity and are therefore easily traceable.

Unicast RPF is implemented as follows:

- ICMP packets have no session, so each packet is checked.
- UDP and TCP have sessions, so the initial packet requires a reverse route lookup. Subsequent packets arriving during the session are checked using an existing state maintained as part of the session. Non-initial packets are checked to ensure they arrived on the same interface used by the initial packet.

Note Before using this command, add static **route** command statements for every network that can be accessed on the interfaces you wish to protect. Only enable this command if routing is fully specified. Otherwise, PIX Firewall will stop traffic on the interface you specify if routing is not in place.

Use the **show interface** command to view the number dropped packets, which appears in the "unicast rpf drops" counter.

Examples

The following example protects traffic between the inside and outside interfaces and provides **route** command statements for two networks, 10.1.2.0 and 10.1.3.0, that connect to the inside interface via a hub:

```
ip address inside 10.1.1.1 255.255.0.0
route inside 10.1.2.0 255.255.0.0 10.1.1.1 1
route inside 10.1.3.0 255.255.0.0 10.1.1.1 1
ip verify reverse-path interface outside
ip verify reverse-path interface inside
```

The **ip verify reverse-path interface outside** command statement protects the outside interface from network ingress attacks from the Internet, whereas the **ip verify reverse-path interface inside** command statement protects the inside interface from network egress attacks from users on the internal network.

The following is sample output from the **show ip verify statistics** and **clear ip verify statistics** commands:

```
pixfirewall(config)# show ip verify statistics
interface outside: 2 unicast rpf drops
interface inside: 1 unicast rpf drops
interface intf2: 3 unicast rpf drops

pixfirewall(config)# clear ip verify statistics

pixfirewall(config)# show ip verify statistics
```

```
interface outside: 0 unicast rpf drops
interface inside: 0 unicast rpf drops
interface intf2: 0 unicast rpf drops
```

isakmp

Configures the Internet Security Association Key Management Protocol (ISAKMP) for IPSec Internet Key Exchange (IKE). See also the **isakmp policy** command.

[**no**] **isakmp client configuration address-pool local** *pool-name* [*interface-name*]

[**no**] **isakmp enable** *interface-name*

[**no**] **isakmp identity** {**address | hostname |** [**key-id** *key_id_string*]}

isakmp keepalive *seconds* [*retry_seconds*]

[**no**] **isakmp key** *keystring* **address** *peer-address* [**netmask** *mask*] [**no-xauth**] [**no-config-mode**]

isakmp log <#events>

isakmp nat-traversal [*natkeepalive*]

[**no**] **isakmp peer fqdn** *fqdn* **no-xauth no-config-mode**

clear [**crypto**] **isakmp sa**

clear isakmp

show isakmp identity

show isakmp sa [**detail**]

Syntax Description

address	The IP address of the host exchanging ISAKMP identity information.
fqdn *fqdn*	The fully qualified domain name of the peer. This is used to identify a peer that is a security gateway.
hostname	The name of the host exchanging ISAKMP identity information.
interface-name	The name of the interface on which to enable ISAKMP negotiation.
keepalive *seconds*	The keepalive interval can be between 10 and 3600 seconds. The retry interval can be between 2 and 10 seconds, with the default being 2 seconds. The retry interval is the interval between retries after a keepalive response has not been received. You can specify the keepalive interval without specifying the retry interval, but cannot specify the retry interval without specifying the keepalive interval.
key	Specifies the authentication pre-shared key. Use any combination of alphanumeric characters up to 128 bytes. This pre-shared key must be identical at both peers.
key-id *key_id_string*	String used by the remote peer to look up the pre-shared key. (This is intended for use with third-party VPN headend devices that do not support the Unity protocol.)
log	Sets the size of the log buffer.
<#events>	(Min 0, max 50,000, default 0 (disabled); each event uses 20 bytes)

netmask *mask*	(Optional) The netmask of 0.0.0.0. can be entered as a wildcard indicating the key could be used for any peer that does not have a key associated with its specific IP address.
natkeepalive	Sets the NAT keep alive interval, from 10 to 3600 seconds. The default is 20 seconds.
nat-traversal	Turns on or off NAT traversal. (NAT traversal is off by default.)
no-config-mode	This is only to be used if you enabled the IKE Mode Configuration feature, and you have an IPSec peer that is a gateway. This option associates a given pre-shared key with a gateway and allows an exception to the IKE Mode Configuration feature enabled by the **crypto map client configuration address** command.
no-xauth	This is only to be used if you enabled the Xauth feature, and you have an IPSec peer that is a gateway. This option associates a given pre-shared key with a gateway and allows an exception to the Xauth feature enabled by the **crypto map client authentication** command.
peer-address	Specifies the IPSec peer's IP address for the pre-shared key.
pool-name	Specify the name of a local address pool to allocate the dynamic client IP.
retry_seconds	Specifies the time interval before a keepalive message is sent if a keepalive response is not received from the previous request.

Command Modes

Configuration mode.

Defaults

By default, NAT traversal (**isakmp nat-traversal**) is disabled.

The default ISAKMP identity is **isakmp identity hostname**.

Usage Guidelines

The **show isakmp identity** command displays the current ISAKMP identity.

The **show isakmp sa** command displays all current IKE security associations between the PIX Firewall and its peer.

The sections that follow describe each **isakmp** command.

isakmp client configuration address-pool local

The **isakmp client configuration address-pool local** command is used to configure the IP address local pool to reference IKE. Use the **no crypto isakmp client configuration address-pool local** command to restore to the default value.

Before using this command, use the **ip local pool** command to define a pool of local addresses to be assigned to a remote IPSec peer.

Examples

The following example references IP address local pools to IKE with "mypool" as the pool-name:

```
isakmp client configuration address-pool local mypool outside
```

isakmp enable

Use the **isakmp enable** *interface-name* command to enable ISAKMP negotiation on the interface on which the IPSec peer will communicate with the PIX Firewall. Use the **no isakmp enable** command to disable IKE.

The following example shows how to disable IKE on the inside interface:

```
no isakmp enable inside
```

isakmp identity

To define the ISAKMP identity the PIX Firewall uses when participating in the IKE protocol, use the **isakmp identity** command. Use **no isakmp identity** command to reset the ISAKMP identity to the default value of IP address. The default ISAKMP identity is **hostname**.

When two peers use IKE to establish IPSec security associations, each peer sends its ISAKMP identity to the remote peer. It will send either its IP address or host name depending on how each has its ISAKMP identity set. By default, the PIX Firewall unit's ISAKMP identity is set to the IP address. As a general rule, set the PIX Firewall and its peer's identities in the same way to avoid an IKE negotiation failure. This failure could be due to either the PIX Firewall or its peer not recognizing its peer's identity.

Note If you are using RSA signatures as your authentication method in your IKE policies, we recommend that you set each participating peer's identity to **hostname**. Otherwise, the ISAKMP security association to be established during Phase 1 of IKE may fail.

The following example uses pre-shared keys between the two PIX Firewall units (PIX Firewall 1 and PIX Firewall 2) that are peers, and sets both their ISAKMP identities to host name.

At the PIX Firewall 1, the ISAKMP identity is set to **hostname**:

```
isakmp identity hostname
```

At the PIX Firewall 2, the ISAKMP identity is set to **hostname**:

```
isakmp identity hostname
```

isakmp identity key-id

The **isakmp identity key-id** *key_id_string* command sends the specified *key_id_string* using aggressive mode. This is intended to enable third-party VPN headend devices that do not support the Unity protocol to interoperate with a DHCP-enabled firewall at a remote site.

Note If the VPN client feature is enabled on the firewall, the **vpnclient** group name takes precedence over the **isakmp identity key-id** setting, and the firewall sends **vpnclient** group name as the **key-id**.

isakmp keepalive

The **isakmp keepalive** *seconds* [*retry_seconds*] command sets the keepalive lifetime interval. The keepalive interval can be between 10 and 3600 seconds. The retry interval can be between 2 and 10 seconds, with the default being 2 seconds. The retry interval is the interval between retries after a keepalive response has not been received. You can specify the keepalive lifetime interval without specifying the retry interval, but cannot specify the retry interval without specifying the keepalive lifetime interval.

isakmp key address

To configure a pre-shared authentication key and associate the key with an IPSec peer address or host name, use the **isakmp key address** command. Use the **no isakmp key address** command to delete a pre-shared authentication key and its associated IPSec peer address.

You would configure the pre-shared key at both peers whenever you specify pre-shared key in an IKE policy. Otherwise, the policy cannot be used because it will not be submitted for matching by the IKE process.

A netmask of 0.0.0.0. can be entered as a wildcard indicating that any IPSec peer with a given valid pre-shared key is a valid peer.

Note The PIX Firewall or any IPSec peer can use the same authentication key with multiple peers, but this is not as secure as using a unique authentication key between each pair of peers.

Configure a pre-shared key associated with a given security gateway to be distinct from a wildcard, pre-shared key (pre-shared key plus a netmask of 0.0.0.0) used to identify and authenticate the remote VPN clients.

The no-xauth or no-config-mode command options are to be used only if the following criteria are met:

- You are using the pre-shared key authentication method within your IKE policy.
- The security gateway and VPN client peers terminate on the same interface.
- The Xauth or IKE Mode Configuration feature is enabled for VPN client peers.

The **isakmp key** *keystring* **address** *ip-address* [**no-xauth**] [**no-config-mode**] command lets you configure a pre-shared authentication key, associate the key with a given security gateway's address, and make an exception to the enabled Xauth feature, IKE Mode Configuration feature, or both (the most common case) for this peer.

Both the Xauth and IKE Mode Configuration features are specifically designed for remote VPN clients. The Xauth feature allows the PIX Firewall to challenge the peer for a username and password during IKE negotiation. The IKE Mode Configuration enables the PIX Firewall to download an IP address to the peer for dynamic IP address assignment. Most security gateways do not support the Xauth and IKE Mode Configuration features.

You cannot enable Xauth or IKE Mode Configuration on a interface when terminating an L2TP/IPSec tunnel using the Microsoft L2TP/IPSec client v1.0 (which is available on Windows NT, Windows XP, Windows 98 and Windows ME OS). Instead, you can do either of the following:

- Use a Windows 2000 L2TP/IPSec client, or
- Use the **isakmp key** *keystring* **address** *ip-address* **netmask** *mask* **no-xauth no-config-mode** command to exempt the L2TP client from Xauth and IKE Mode Configuration. However, if you exempt the L2TP client from Xauth or IKE Mode Configuration, all the L2TP clients must be grouped with the same ISAKMP pre-shared key or certificate and have the same fully qualified domain name.

If you have the **no-xauth** command option configured, the PIX Firewall will not challenge the peer for a username and password. Similarly, if you have the **no-config-mode** command option configured, the PIX Firewall will not attempt to download an IP address to the peer for dynamic IP address assignment.

Use the **no key** *keystring* **address** *ip-address* [**no-xauth**] [**no-config-mode**] command to disable the **key** *keystring* **address** *ip-address* [**no-xauth**] [**no-config-mode**] command that you previously enabled.

See the **crypto map client authentication** command within the **crypto map** command page for more information about the Xauth feature. See the **crypto map client configuration address** command within the **crypto map** command page for more information about the IKE Mode Config feature.

The following example shows "sharedkeystring" as the authentication key to share between the PIX Firewall and its peer specified by an IP address of 10.1.0.0:

```
isakmp key sharedkeystring address 10.1.0.0
```

The following example shows use of a wildcard, pre-shared key. The "sharedkeystring" is the authentication key to share between the PIX Firewall and its peer (in this case a VPN client) specified by an IP address of 0.0.0.0. and a netmask of 0.0.0.0.

```
isakmp key sharedkeystring address 0.0.0.0 netmask 0.0.0.0
```

The following example shows use of the command options **no-xauth and no-config-mode** in relation to three PIX Firewall peers that are security gateways. These security gateways terminate IPSec on the same interface as the VPN clients. Both the Xauth and IKE Mode Config features are enabled. This means there is a need to make an exception to these two features for each security gateway. The example shows each security gateway peer has a unique pre-shared key to share with the PIX Firewall. The peers' IP addresses are 10.1.1.1, 10.1.1.2, 10.1.1.3, and the netmask of 255.255.255.255 is specified.

```
isakmp key secretkey1234 address 10.1.1.1 netmask 255.255.255.255 no-xauth no-config-mode
isakmp key secretkey4567 address 10.1.1.2 netmask 255.255.255.255 no-xauth no-config-mode
isakmp key secretkey7890 address 10.1.1.3 netmask 255.255.255.255 no-xauth no-config-mode
```

isakmp log <#events>

A circular event tracing buffer has been added to assist in troubleshooting when syslogs are unavailable. The event buffer is disabled by default. After it is enabled, the following events will be recorded in the buffer.

```
DPD_TX
DPD_RX
DPD_ACK_TX
DPD_ACK_RX
DPD_FAIL
P1_RETRAN
P2_RETRAN
VPNC_CONNECT
VPNC_DISCONNECT
IKE_DELETE_TX
IKE_DELETE_RX
MALFORMED_PAYLOAD
DUPLICATE_PACKET
P1_INIT
P1_RESP
P1_DONE
P2_INIT
P2_RESP
P2_DONE
P1_REKEY
```

```
mypix# show isakmp log


    16:18:31.964 UTC Fri May 21 2004, peer 63.67.72.161, P2_INIT
    16:18:31.774 UTC Fri May 21 2004, peer 63.67.72.161, VPNC_CONNECT
    16:18:31.774 UTC Fri May 21 2004, peer 63.67.72.161, P1_DONE
    16:18:30.234 UTC Fri May 21 2004, peer 63.67.72.161, P1_INIT
    02:11:23.762 UTC Fri May 21 2004, peer 63.67.72.161, IKE_DELETE_TX
    02:11:22.982 UTC Fri May 21 2004, peer 63.67.72.161, P1_RESP
    02:11:22.762 UTC Fri May 21 2004, peer 63.67.72.161, VPNC_DISCONNECT
    02:11:22.692 UTC Fri May 21 2004, peer 63.67.72.161, P2_RETRAN (1)
    02:11:22.682 UTC Fri May 21 2004, peer 63.67.72.161, P2_RETRAN (1)
    02:11:22.592 UTC Fri May 21 2004, peer 63.67.72.161, P2_RETRAN (1)
    02:11:21.702 UTC Fri May 21 2004, peer 63.67.72.161, P2_DONE
```

```
02:11:21.682 UTC Fri May 21 2004, peer 63.67.72.161, P2_DONE
02:11:21.432 UTC Fri May 21 2004, peer 63.67.72.161, P2_RESP
02:11:21.422 UTC Fri May 21 2004, peer 63.67.72.161, P2_RESP
02:11:18.782 UTC Fri May 21 2004, peer 63.67.72.161, DPD_TX (1074122247)
02:11:16.701 UTC Fri May 21 2004, peer 63.67.72.161, P2_DONE
02:11:16.321 UTC Fri May 21 2004, peer 63.67.72.161, P2_RESP
02:11:13.781 UTC Fri May 21 2004, peer 63.67.72.161, DPD_TX (1074122246)
02:11:13.141 UTC Fri May 21 2004, peer 63.67.72.161, P2_RESP
```

isakmp nat-traversal

Network Address Translation (NAT), including Port Address Translation (PAT), is used in many networks where IPSec is also used, but there are a number of incompatibilities that prevent IPSec packets from successfully traversing NAT devices. NAT traversal enables ESP packets to pass through one or more NAT devices.

The firewall supports NAT traversal as described by Version 2 and Version 3 of the IETF "UDP Encapsulation of IPsec Packets" draft, available at http://www.ietf.org/html.charters/ipsec-charter.html, and NAT traversal is supported for both dynamic and static crypto maps. NAT traversal is disabled by default on the firewall.

To enable NAT traversal, check that ISAKMP is enabled (you can enable it with the **isakmp enable** *if_name* command) and then use the **isakmp nat-traversal** [*natkeepalive*] command. (This command appears in the configuration if both ISAKMP is enabled and NAT traversal is enabled.) If you have enabled NAT traversal, you can disable it with the **no isakmp nat-traversal** command. Valid values for *natkeepalive* are from 10 to 3600 seconds. The default is 20 seconds.

If needed, the **show isakmp sa detail** command assists in debugging NAT traversal.

isakmp peer fqdn no-xauth | no-config-mode

The **isakmp peer fqdn** *fqdn* **no-xauth | no-config-mode** command is to be used only if the following criteria are met:

- You are using the RSA signatures authentication method within your IKE policy.
- The security gateway and VPN client peers terminate on the same interface.
- The Xauth or IKE Mode Configuration feature is enabled for VPN client peers.

The **isakmp peer fqdn** *fqdn* **no-xauth | no-config-mode** command lets you identify a peer that is a security gateway and make an exception to the enabled Xauth feature, IKE Mode Configuration feature, or both (the most common case) for this peer.

Both the Xauth and IKE Mode Configuration features are specifically designed for remote VPN clients. The Xauth feature allows the PIX Firewall to challenge the peer for a username and password during IKE negotiation. The IKE Mode Configuration feature enables the PIX Firewall to download an IP address to the peer for dynamic IP address assignment. Most security gateways do not support the Xauth and IKE Mode Configuration features.

If you have the **no-xauth** command option configured, the PIX Firewall will not challenge the peer for a username and password. Similarly, if you have the **no-config-mode** command option configured, the PIX Firewall will not attempt to download an IP address to the peer for dynamic IP address assignment.

Note If you are using RSA signatures as your authentication method in your IKE policies, we recommend that you set each participating peer's identity to hostname using the **isakmp identity hostname** command. Otherwise, the ISAKMP security association to be established during Phase 1 of IKE may fail.

Use the **no isakmp peer fqdn** *fqdn* **no-xauth | no-config-mode** command to disable the **isakmp peer fqdn** *fqdn* **no-xauth | no-config-mode** command that you previously enabled.

See the **crypto map client authentication** within the **crypto map** command page for more information about the Xauth feature. See the **crypto map client configuration address** command within the **crypto map** command page for more information about the IKE Mode Config feature.

The following example shows use of the command options **no-xauth and no-config-mode** in relation to three PIX Firewall peers that are security gateways. These security gateways terminate IPSec on the same interface as the VPN clients. Both the Xauth and IKE Mode Config features are enabled. This means there is a need to make an exception to these two features for each security gateway. Each security gateway peer's fully qualified domain name is specified.

```
isakmp peer fqdn hostname1.example.com no-xauth no-config-mode
isakmp peer fqdn hostname2.example.com no-xauth no-config-mode
isakmp peer fqdn hostname3.example.com no-xauth no-config-mode
```

show isakmp sa

To view all current IKE security associations between the PIX Firewall and its peer, use the **show isakmp sa** command.

The following is sample output from the **show isakmp sa** command after IKE negotiations were successfully completed between the PIX Firewall and its peer:

```
pixfirewall# show isakmp sa
    dst src statependingcreated
    16.132.40.216.132.30.2QM_IDLE01
```

The following is sample output from the **show isakmp sa detail** command (used for debugging NAT traversal):

```
pixfirewall# show isakmp sa detail
Total     : 1
Embryonic : 0
         Local            Remote          Encr  Hash    Auth  State      Lifetime
   192.168.10.2:4500    192.168.10.5:1178 3des  sha     psk  QM_IDLE       117
```

`Local` is the IP address and port of the firewall on which the command is run (the format is IP_Address:port); `Remote` is the peer IP address and port; `Encr` is the encryption algorithm; `Hash` is the hash algorithm; `Auth` is the authorization method, preshared key, or rsa; `State` is the state of the connection, and `Lifetime` is either the time until re-key or until expiration and deletion.

clear isakmp

The **clear isakmp** command removes all **isakmp** command statements from the configuration.

clear [crypto] isakmp sa

The **clear** [**crypto**] **isakmp sa** command deletes active IKE security associations. The keyword **crypto** is optional.

isakmp policy

Configures specific Internet Key Exchange (IKE) algorithms and parameters, within the IPSec Internet Security Association Key Management Protocol (ISAKMP) framework, for the Authentication Header (AH) and Encapsulating Security Payload (ESP) IPSec protocols. See also the **isakmp** command.

[**no**] **isakmp policy** *priority* **authentication** *pre-share* | *rsa-sig*

[**no**] **isakmp policy** *priority* **encryption aes | aes-192| aes-256 | des | 3des**

[**no**] **isakmp policy** *priority* **group 1 | 2 | 5**

[**no**] **isakmp policy** *priority* **hash** *md5* | *sha*

[**no**] **isakmp policy** *priority* **lifetime** *seconds*

show isakmp policy

Syntax Description

3des	Specifies that the Triple DES encryption algorithm is to be used in the IKE policy.
aes	Selecting this option means that encrypted IKE messages protected by this suite are encrypted using AES with a 128-bit key.
aes-192	Selecting this option means that encrypted IKE messages protected by this suite are encrypted using AES with a 192-bit key.
aes-256	Selecting this option means that encrypted IKE messages protected by this suite are encrypted using AES with a 256-bit key.
des	Specifies 56-bit DES-CBC as the encryption algorithm to be used in the IKE policy.
group 1	Specify that the 768-bit Diffie-Hellman group is to be used in the IKE policy. This is the default value.
group 2	Specifies that the 1024-bit Diffie-Hellman group 2 be used in the IKE policy.
group 5	Specifies that the 1536-bit Diffie-Hellman group 5 be used in the IKE policy.
lifetime *seconds*	Specify how many seconds each security association should exist before expiring. To propose a finite lifetime, use an integer from 120 to 86,400 seconds (one day). Specify 0 seconds for infinite lifetime.
md5	Specify MD5 (HMAC variant) as the hash algorithm to be used in the IKE policy.
pre-share	Specify pre-shared keys as the authentication method.
priority	Uniquely identifies the Internet Key Exchange (IKE) policy and assigns a priority to the policy. Use an integer from 1 to 65,534, with 1 being the highest priority and 65,534 the lowest.
rsa-sig	Specify RSA signatures as the authentication method. RSA signatures provide non-repudiation for the IKE negotiation. This basically means you can prove to a third party whether you had an IKE negotiation with the peer.
sha	Specify SHA-1 (HMAC variant) as the hash algorithm to be used in the IKE policy. This is the default hash algorithm.

Command Modes Configuration mode.

Defaults The default ISAKMP policy encryption is **des**.

The default hash algorithm is SHA-1 (HMAC variant).

Usage Guidelines The **isakmp policy** command lets you negotiate IPSec security associations and enable IPSec secure communications.

The following is an example of the **isakmp policy** command:

```
isakmp policy 93 group 2
```

Note The Cisco VPN Client Version 3.x requires **isakmp policy** to have DH **group 2** configured. (If you have DH **group 1** configured, the Cisco VPN Client cannot connect.)

AES support is available on firewalls licensed for VPN-3DES only. Due to the large key sizes provided by AES, ISAKMP negotiation should use Diffie-Hellman (DH) **group 5** instead of **group 1** or **group 2**. This is done with the **isakmp policy priority group 5** command.

The **show isakmp policy** command displays parameters for each IKE policy, including defaults.

isakmp policy authentication

The isakmp policy authentication command lets you specify the authentication method within an IKE policy. IKE policies define a set of parameters to be used during IKE negotiation.

If you specify RSA signatures, you must configure the PIX Firewall and its peer to obtain certificates from a CA. If you specify pre-shared keys, you must separately configure these pre-shared keys within the PIX Firewall and its peer.

Use the **no isakmp policy authentication** command to reset the authentication method to the default value of RSA signatures.

The following example shows use of the **isakmp policy authentication** command. This example sets the authentication method of rsa-signatures to be used within the IKE policy with the priority number of 40.

```
isakmp policy 40 authentication rsa-sig
```

isakmp policy encryption

To specify the encryption algorithm to be used within an IKE policy, use the **isakmp policy encryption** command. AES with a 128-bit key (**aes**), AES with a 192-bit key (**aes-192**), AES with a 256-bit key (**aes-256**), DES (**des**), and 3DES (**3des**) are the supported encryption algorithms. (IKE policies define the set of parameters to be used during IKE negotiation.)

Use the **no isakmp policy encryption** command to reset the encryption algorithm to the default value, which is **des**.

The following example shows use of the **isakmp policy encryption** command; it sets 128-bit key AES encryption as the algorithm to be used within the IKE policy with the priority number of 25.

```
isakmp policy 25 encryption aes
```

The following example sets the 3DES algorithm to be used within the IKE policy with the priority number of 40.

```
isakmp policy 40 encryption 3des
```

isakmp policy group

Use the **isakmp policy group** command to specify the Diffie-Hellman group to be used in an IKE policy. IKE policies define a set of parameters to be used during IKE negotiation.

There are three group options: 768-bit (DH Group 1), 1024-bit (DH Group 2), or 1536-bit (DH Group 5). The 1024-bit and 1536-bit Diffie-Hellman Groups provide stronger security, but it require more CPU time to execute.

Use the **no isakmp policy group** command to reset the Diffie-Hellman group identifier to the default value of group 1 (768-bit Diffie Hellman).

The following example shows use of the **isakmp policy group** command. This example sets group 2, the 1024-bit Diffie Hellman, to be used within the IKE policy with the priority number of 40.

```
isakmp policy 40 group 2
```

Note Cisco VPN Client Version 3.x uses Diffie-Hellman group 2 and Cisco VPN Client 3000 Version 2.5/2.6 uses Diffie-Hellman group 1. If you are using Cisco VPN Client Version 3.x, configure Diffie-Hellman group 2 by using the **isakmp policy group 2** command.

isakmp policy hash

Use the **isakmp policy hash** command to specify the hash algorithm to be used in an IKE policy. IKE policies define a set of parameters to be used during IKE negotiation.

There are two hash algorithm options: SHA-1 and MD5. MD5 has a smaller digest and is considered to be slightly faster than SHA-1.

To reset the hash algorithm to the default value of SHA-1, use the **no isakmp policy hash** command.

The following example shows use of the **isakmp policy hash** command. This example sets the MD5 hash algorithm to be used within the IKE policy with the priority number of 40.

```
isakmp policy 40 hash md5
```

isakmp policy lifetime

To specify the lifetime of an IKE security association before it expires, use the **isakmp policy lifetime** command. An infinite lifetime can also be specified in case the peer does not propose a lifetime. Use the **no isakmp policy lifetime** command to reset the security association lifetime to the default value of 86,400 seconds (one day).

When IKE begins negotiations, it looks to agree upon the security parameters for its own session. The agreed-upon parameters are then referenced by a security association at each peer. The security association is retained by each peer until the security association's lifetime expires. Before a security association expires, it can be reused by subsequent IKE negotiations, which can save time when setting up new IPSec security associations. New security associations are negotiated before current security associations expire.

To save setup time for IPSec, configure a longer IKE security association lifetime. However, the shorter the lifetime (up to a point), the more secure the IKE negotiation is likely to be.

Note If the IKE security association is set to an infinite lifetime, but the peer proposes a finite lifetime, then the negotiated finite lifetime from the peer will be used.

Note When PIX Firewall initiates an IKE negotiation between itself and an IPSec peer, an IKE policy can be selected only if the lifetime of the peer's policy is shorter than or equal to the lifetime of its policy. Then, if the lifetimes are not equal, the shorter lifetime will be selected.

The following example shows use of the **isakmp policy lifetime** command. This example sets the lifetime of the IKE security association to 50,400 seconds (14 hours) within the IKE policy with the priority number of 40.

```
isakmp policy 40 lifetime 50400
```

The following example sets the IKE security association to an infinite lifetime.

```
isakmp policy 40 lifetime 0
```

show isakmp policy

To view the parameters for each IKE policy including the default parameters, use th**e show isakmp policy** command.

The following is sample output from the **show isakmp policy** command after two IKE policies were configured (with priorities 70 and 90 respectively):

```
show isakmp policy

Protection suite priority 70
        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 90
        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 Although the output shows "no volume limit" for the lifetimes, you can currently only configure a time lifetime (such as 86,400 seconds) or infinity; volume limit lifetimes are not currently configurable.

Examples

The following is sample output from the **show isakmp** and **show isakmp policy** commands for a configuration using Diffie-Hellman group 5 in its ISAKMP policy:

```
pixfirewall(config)# show isakmp
isakmp enable outside
isakmp key ******** address 0.0.0.0 netmask 0.0.0.0
isakmp policy 1 authentication pre-share
isakmp policy 1 encryption 3des
isakmp policy 1 hash md5
isakmp policy 1 group 5
isakmp policy 1 lifetime 86400

pixfirewall(config)# show isakmp policy
Protection suite of priority 8
        encryption algorithm:   Three key triple DES
        hash algorithm:         Message Digest 5
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #5 (1536 bit)
        lifetime:               86400 seconds, no volume limit
```

Related Commands

ca	For Certificate Enrollment Protocol (CEP), creates and enrolls RSA key pairs into a Public Key Infrastructure (PKI).
crypto dynamic-map	Configures the IPSec crypto dynamic-map policy.
crypto ipsec	Configures the transform set and IPSec security association (SA) lifetime.
crypto map	Configures the IPsec crypto map policy.

kill

Terminate a Telnet session.

kill *telnet_id*

Syntax Description

telnet_id	Telnet session ID.

Command Modes

Privileged mode.

Usage Guidelines

The **kill** command terminates a Telnet session. Use the **who** command to view the Telnet session ID value. When you kill a Telnet session, the PIX Firewall lets any active commands terminate and then drops the connection without warning the user.

Examples

The following is sample output from the **show who** command, which is used to list the active Telnet sessions, and the use of the **kill** command to end Telnet session 2:

```
show who
2: From 10.10.54.0
kill 2
```

Related Commands		
	who	Shows the active administration sessions on the firewall.
	telnet	Adds Telnet access to the firewall console and sets the idle timeout.

logging

Enable or disable syslog and SNMP logging.

[**no**] **logging on**

[**no**] **logging buffered** *level*

[**no**] **logging console** *level*

logging device-id {**hostname** | **ipaddress** *if_name* | **string** *text*}

no logging device-id

[**no**] **logging facility** *facility*

[**no**] **logging history** *level*

[**no**] **logging host** [*in_if_name*] *ip_address* [*protocol*/*port*] [**format emblem**]

[**no**] **logging message** *syslog_id* [**level** *level*]

[**no**] **logging monitor** *level*

[**no**] **logging queue** *queue_size*

[**no**] **logging standby**

[**no**] **logging timestamp**

[**no**] **logging trap** *level*

clear logging [**disable**]

show logging [**message** {*syslog_id* | **all**} | **level** | **disabled**]

show logging queue

Syntax Description		
	all	All syslog message IDs.
	buffered	Send syslog messages to an internal buffer that can be viewed with the **show logging** command. Use the **clear logging** command to clear the message buffer. New messages append to the end of the buffer.
	clear	Clear the buffer for use with the **logging buffered** command.

console	Specify that syslog messages appear on the PIX Firewall console as each message occurs. You can limit the types of messages that appear on the console with *level*. We recommend that you do not use this command in production mode because its use degrades PIX Firewall performance.
device-id	The device ID of the PIX Firewall to include in the syslog message.
disabled	Clear or display suppressed messages. You can suppress messages with the **no logging message** command.
facility	Specify the syslog facility. The default is 20.
facility	Eight facilities LOCAL0(16) through LOCAL7(23); the default is LOCAL4(20). Hosts file the messages based on the *facility* number in the message.
format emblem	This option enables EMBLEM format logging on a per-syslog-server basis. EMBLEM format logging is available for UDP syslog messages only and is disabled by default.
history	Set the SNMP message level for sending syslog traps.
host	Specify a syslog server that will receive the messages sent from the PIX Firewall. You can use multiple **logging host** commands to specify additional servers that would all receive the syslog messages. A PIX Firewall Syslogs Server can be configured to receive syslogs over UDP or TCP, not both. Likewise the PIX Firewall can send either UDP or TCP syslog messages to the PIX Firewall Syslog Server.
hostname	Specifies to use the host name of the PIX Firewall to uniquely identify the syslog messages from the PIX Firewall.
if_name	Specifies the name of the interface whose IP address is used to uniquely identify the syslog messages from the PIX Firewall.
in_if_name	Interface on which the syslog server resides.
ip_address	Syslog server's IP address.
ipaddress	Specifies to use the IP address of the specified PIX Firewall interface to uniquely identify the syslog messages from the PIX Firewall.
level	Specify the syslog message level as a number or string. The *level* you specify means that you want that *level* and those less than the *level*. For example, if *level* is **3**, syslog displays **0**, **1**, **2**, and **3** messages**.** Possible number and string *level* values are: • **0**—**emergencies**—System unusable messages • **1**—**alerts**—Take immediate action • **2**—**critical**—Critical condition • **3**—**errors**—Error message • **4**—**warnings**—Warning message • **5**—**notifications**—Normal but significant condition • **6**—**informational**—Information message • **7**—**debugging**—Debug messages and log FTP commands and WWW URLs

message	Specify a message to be allowed. Use the **no logging message** command to suppress a syslog message. Use the **clear logging disabled** command to reset the disallowed messages to the original set. Use the **show message disabled** command to list the suppressed messages. All syslog messages are permitted unless explicitly disallowed. The "PIX Startup begin" message cannot be blocked and neither can more than one message per command statement.
monitor	Specify that syslog messages appear on Telnet sessions to the PIX Firewall console.
on	Start sending syslog messages to all output locations. Stop all logging with the **no logging on** command.
port	The port from which the PIX Firewall sends either UDP or TCP syslog messages. This must be same port at which the syslog server listens. For the UDP port, the default is 514 and the allowable range for changing the value is 1025 through 65535. For the TCP port, the default is 1470, and the allowable range is 1025 through 65535. TCP ports only work with the PIX Firewall Syslog Server.
protocol	The protocol over which the syslog message is sent; either **tcp** or **udp**. PIX Firewall only sends TCP syslog messages to the PIX Firewall Syslog Server. You can only view the port and protocol values you previously entered by using the **write terminal** command and finding the command in the listing—the TCP protocol is listed as 6 and the UDP protocol is listed as 17.
queue *queue_size*	Specifies the size of the queue for storing syslog messages. Use this parameter before the syslog messages are processed. The queue parameter defaults to 512 messages, 0 (zero) indicates unlimited (subject to available block memory), and the minimum is one message.
standby	Let the failover standby unit also send syslog messages. This option is disabled by default. You can enable it to ensure that the standby unit's syslog messages stay synchronized should failover occur. However, this option causes twice as much traffic on the syslog server. Disable with the **no logging standby** command.
syslog_id	Specify a message number to disallow or allow. If a message is listed in syslog as %PIX-1-101001, use "101001" as the *syslog_id*. Refer to *Cisco PIX Firewall System Log Messages* for message numbers.
text	Specifies the text string to uniquely identify the syslog messages from the PIX Firewall. The maximum length is 16 characters with no whitespace (blanks) allowed.
timestamp	Specify that syslog messages sent to the syslog server should have a time stamp value on each message.
trap	Set logging level only for syslog messages.

Defaults

EMBLEM format logging is disabled by default.

The **logging device-id** command is disabled by default.

Console logging (the **logging console** command) is disabled by default.

Command Modes

Configuration mode.

Usage Guidelines

The **logging** command lets you enable or disable sending informational messages to the console, to a syslog server, or to an SNMP management station.

The PIX Firewall provides more information in messages sent to a syslog server than at the console, but the console provides enough information to permit effective troubleshooting.

Note Do not use the **logging console** command when the PIX Firewall is in production mode because it degrades system performance. Instead, use the **logging buffered** command to start logging, the **show logging** command to view the messages, and the **clear logging** command to clear the buffer to make viewing the most current messages easier.

The **aaa accounting** **authentication enable console** command causes syslog messages to be sent (at syslog level 4) each time the configuration is changed from the serial console.

The **show logging** command displays which logging options are enabled. If the **logging buffered** command is in use, the **show logging** command lists the current message buffer. The **show logging disabled** command displays suppressed syslog messages.

logging device-id

The **logging device-id** command displays a unique device ID in non-EMBLEM format syslog messages that are sent to the syslog server. This command is available in PIX Firewall software Version 6.2.2.115 and higher.

If enabled, the PIX Firewall displays the device ID in all non-EMBLEM-formatted syslog messages. However, it does not affect the syslog message text that is in EMBLEM format.

Note The device ID part of the syslog message is viewed through the syslog server only and not directly on the firewall.

If the **ipaddress** option is used, the device ID becomes the specified PIX Firewall interface IP address, regardless of the interface from which the message is sent. This provides a single consistent device ID for all messages sent from the device.

logging history

Set the SNMP message level with the **logging history** command.

logging host

The **logging host** *ip_address* **format emblem** command enables EMBLEM format logging on a per-syslog-server basis. EMBLEM format logging is available for UDP syslog messages only (because the RME syslog analyzer only supports UDP syslog messages). If EMBLEM format logging is enabled for a particular syslog host, then EMBLEM format messages are sent to that host. If the **logging timestamp** option is also enabled, then EMBLEM format messages with a time stamp are sent. EMBLEM format logging is disabled by default.

logging message

To change the level of a syslog message, use the **logging message** *syslog_id* **level** *level* command. The **no logging message** command cannot block the "`%PIX-6-199002: PIX startup completed. Beginning operation`." syslog message.

logging queue

The **logging queue** command lets you specify the size of the syslog message queue for the messages waiting to be processed. When traffic is heavy, messages may be discarded.

The **show logging queue** command lists:

- Number of messages in the queue
- Highest number of messages recorded in the queue
- Number of messages discarded because block memory was not available to process them

logging standby

The **logging standby** command lets the failover standby unit send syslog messages. This option is disabled by default. You can enable it to ensure that the standby unit's syslog messages stay synchronized should failover occur. However, this option causes twice as much traffic on the syslog server. Disable with the **no logging standby** command.

logging timestamp

The **logging timestamp** command requires that the **clock** command be set.

logging trap

Set the syslog message level with the **logging trap** command.

Troubleshooting

If you are using TCP as the logging transport protocol, the PIX Firewall stops passing traffic as a security measure if any of the following error conditions occur: the PIX Firewall is unable to reach the syslog server; the syslog server is misconfigured (such as with PFSS, for example); or the disk is full. (UDP-based logging does not prevent the PIX Firewall from passing traffic if the syslog server fails.)

To enable the PIX Firewall to pass traffic again, do the following:

Step 1 Identify and correct the syslog server connectivity, misconfiguration, or disk space error condition.

Step 2 Enter the command **`logging host inside 10.1.1.1 tcp/1468`** to enable the logging again.

Alternately, you can change the logging to default logging on UDP/514 by issuing the command **`logging host inside 10.1.1.1`**. UDP-based logging passes traffic even if the syslog server fails.

For more information

For more information on syslog and the use of the **logging** command, refer to *Cisco PIX Firewall System Log Messages*. You can also use *Cisco PIX Firewall System Log Messages* to get the message numbers that can be individually suppressed with the **logging message** command.

Examples

The following example shows how to start console logging and view the results:

```
pixfirewall(config)# logging buffered debugging
pixfirewall(config)# show logging
Syslog logging: enabled
    Timestamp logging: disabled
    Console logging: disabled
    Monitor logging: disabled
    Buffer logging: level debugging, 37 messages logged
    Trap logging: disabled
```

```
305001: Portmapped translation built for gaddr 209.165.201.5/0 laddr 192.168.1.2/256
...
```

The line of output starting with 305001 shows a translation to a PAT global through global address 209.165.201.5 from a host at 192.168.1.2. The "305001" identifies a syslog message for creating a translation through a PAT global. Refer to *Cisco PIX Firewall System Log Messages* for more information on syslog messages.

The following is sample output from the **show logging** command with the **logging device-id hostname** command configured on a host named **pixfirewall-1** (notice the last line):

```
pixfirewall-1(config)# logging device-id hostname
pixfirewall-1(config)# show logging
Syslog logging: disabled
Facility: 20
Timestamp logging: disabled
Standby logging: disabled
Console logging: level debugging, 0 messages logged
Monitor logging: level debugging, 0 messages logged
Buffer logging: disabled
Trap logging: disabled
History logging: disabled
Device ID: hostname "pixfirewall-1"
```

The next example lists the output of the **logging queue** and **show logging queue** commands:

```
pixfirewall(config)# logging queue 0
pixfirewall(config)# show logging queue
Logging Queue length limit : Unlimited
Current 5 msg on queue, 3513 msgs most on queue, 1 msg discard.
```

In this example, the **logging queue** command is set to 0, which means you want an unlimited number of messages; in other words, all syslog messages, to be processed. The **show logging queue** command shows that 5 messages are queued, 3513 messages was the greatest number of messages in the queue at one time since the PIX Firewall was last booted, and that 1 message was discarded. Even though set for unlimited, should the amount of block memory be exhausted, messages can still be discarded.

The following is sample output from the **show logging** command output when the TCP syslog server is unreachable. Consequently, the PIX Firewall stops passing traffic and logging to the inside is set as **disabled**:

```
pixfirewall(config)# show logging
Syslog logging: enabled
Timestamp logging: enabled
Standby logging: disabled
Console logging: disabled
Monitor logging: disabled
Buffer logging: level debugging, 827 messages logged
Trap logging: level debugging, facility 20, 840 messages logged
   Logging to inside 10.1.1.1 tcp/1468 disabled
```

The following examples show how to change the level of a syslog message and display its current and default level:

```
pixfirewall(config)# logging message 403503
pixfirewall(config)# show logging message 403503
syslog 403503: default-level errors (enabled)

pixfirewall(config)# logging message 403503 level 1
pixfirewall(config)# show logging message 403503
syslog 403503: default-level errors, current-level alerts (enabled)

pixfirewall(config)# logging message 403503 level 6
```

```
pixfirewall(config)# show logging message 403503
syslog 403503: default-level errors, current-level informational (enabled)

pixfirewall(config)# logging message 403503 level 3
pixfirewall(config)# show logging message 403503
syslog 403503: default-level errors (enabled)
```

Related Commands

auto-update	Configures auto update support.
telnet	Adds Telnet access to the firewall console and sets the idle timeout.
terminal	Sets terminal line parameters.

login

Initiates the log-in prompt on the PIX Firewall for starting a session, accessing another privilege level, or command mode as a specific user.

login

Syntax Description

login	Specifies to log in as a particular user.

Command Modes

Unprivileged mode.

Usage Guidelines

The **login** command logs the user into the PIX Firewall, another privilege level, or command mode using the local user authentication database created with the **username** command. This command is available in unprivileged mode.

A user who has logged in can use the **logout**, **exit**, or **quit** commands to go back to unprivileged mode.

Examples

The following example shows the prompt after you enter the **login** command:

```
pixfirewall> login
Username:
```

Related Commands

privilege	Configures privilege levels for commands.
username	Configures the local user authentication database.

CHAPTER 7

M through R Commands

mac-list

Adds a list of MAC addresses using a first match search. This command is used by the firewall VPN client in performing MAC-based authentication.

[**no**] **mac-list** *id* **deny**|**permit** *mac macmask*

show mac-list [*id*]

clear mac-list [*id*]

Syntax Description

deny	Traffic matching **deny** is not included in the MAC list and is subjected to both authentication and authorization.
id	MAC access list number.
mac	Source MAC address in *aabbcc.ddeeff.gghhii* form.
macmask	Applies the netmask to *mac*, which is a string of 1's followed by 0's in the form *aabbcc.ddeeff.gghhii*, and allows the grouping of MAC addresses.
permit	Traffic matching **permit** is included in the MAC list and is exempt from authentication and authorization.

Defaults

None.

Command Modes

The **mac-list** command is available in configuration mode.

The **show mac-list** command is available in privileged mode.

Usage Guidelines

The **mac-list** command, similar to the **access-list** command, can be entered multiple times with same *id* to group a set of MAC addresses.

Only AAA exemption is provided. Authorization is automatically exempted for MACs for which authentication is exempted. Other types of AAA with **mac-list** are not supported.

The **clear aaa** command removes the **mac-list** command statements along with the rest of the AAA configuration.

The **show aaa** command displays **mac-list** command statements as part of the AAA configuration.

Note When configuring **mac-exempt**, do not use the same IP address for two MACs. If a **mac-exempt** command is configured for two MACs, M1 and M2, and both attempt to use the same ip address, only the traffic from M1 would be permitted. If a **mac-exempt** is configured for M1 or M2, or if one of them is not configured at all, then the traffic from second host would be allowed to pass. A syslog alerting you to a possible spoof attack, is generated.

Examples

The following example shows how to configure a MAC access list:

```
pixfirewall(config)# mac-list adc permit 00a0.c95d.0282 ffff.ffff.ffff
pixfirewall(config)# mac-list adc deny 00a1.c95d.0282 ffff.ffff.ffff
pixfirewall(config)# mac-list ac permit 0050.54ff.0000 ffff.ffff.0000
pixfirewall(config)# mac-list ac deny 0061.54ff.b440 ffff.ffff.ffff
pixfirewall(config)# mac-list ac deny 0072.54ff.b440 ffff.ffff.ffff

pixfirewall(config)# show mac-list
mac-list adc permit 00a0.c95d.0282 ffff.ffff.ffff
mac-list adc deny 00a1.c95d.0282 ffff.ffff.ffff
mac-list ac permit 0050.54ff.0000 ffff.ffff.0000
mac-list ac deny 0061.54ff.b440 ffff.ffff.ffff
mac-list ac deny 0072.54ff.b440 ffff.ffff.ffff
```

Related Commands

aaa authentication	Enable, disable, or view LOCAL, TACACS+, or RADIUS user authentication on a server designated by the **aaa-server** command, or PDM user authentication.
aaa authorization	Enable or disable LOCAL or TACACS+ user authorization services.
	Exempts a list of MAC addresses from authentication and authorization.
access-list	Create an access list, or use downloadable access lists. (Downloadable access lists are supported for RADIUS servers only.)

management-access

Enables access to an internal management interface on the firewall.

[**no**] **management-access** *mgmt_if*

show management-access

Syntax Description

mgmt_if	The name of the firewall interface to be used as the internal management interface.

Defaults

None.

Command Modes

The **management-access** *mgmt_if* command is available in configuration mode.

The **show management-access** is available in privileged mode.

Usage Guidelines

The **management-access** *mgmt_if* command enables you to define an internal management interface using the IP address of the firewall interface specified in *mgmt_if*. (The firewall interface names are defined by the **nameif** command and displayed in quotes, " ", in the **show interface** output.)

In PIX Firewall software Version 6.3, this command is supported for the following through an IPSec VPN tunnel only, and only one management interface can be defined globally:

- SNMP polls to the *mgmt_if*
- HTTPS requests to the *mgmt_if*
- PDM access to the *mgmt_if*
- Telnet access to the *mgmt_if*
- SSH access to the *mgmt_if*
- Ping to the *mgmt_if*

The **show management-access** command displays the firewall management access configuration.

Examples

The following example shows how to configure a firewall interface named "inside" as the management access interface:

```
pixfirewall(config)# management-access inside
pixfirewall(config)# show management-access
management-access inside
```

mgcp

Configures additional support for the Media Gateway Control Protocol (MGCP) fixup (packet application inspection) and is used with the **fixup protocol mgcp** command.

[**no**] **mgcp call-agent** *ip_address group_id*

[**no**] **mgcp command-queue** *limit*

[**no**] **mgcp gateway** *ip_address group_id*

show mgcp {**commands** | **sessions**} [**detail**]

clear mgcp

Syntax Description

commands	The MGCP commands in the MGCP configuration on the firewall.
group_id	The ID of the Call Agent group, from 0 to 4294967295.
ip_address	The IP address of the gateway.
limit	Maximum number of commands to queue, from 1 to 4294967295.
sessions	The MGCP active sessions.

Defaults

The default for the MGCP command queue is 200.

Command Modes

The **mgcp** command is available in configuration mode.

The **show mgcp** command is available in privileged mode.

Usage Guidelines

The **mgcp** commands are used to provide additional support for the MGCP fixup. The MGCP fixup itself is enabled with the **fixup protocol mgcp** command.

mgcp call-agent

The **mgcp call-agent** command is used to specify a group of Call Agents that can manage one or more gateways. The Call Agent group information is used to open connections for the Call Agents in the group (other than the one a gateway sends a command to) so that any of the Call Agents can send the response. Call Agents with the same *group_id* belong to the same group. A Call Agent may belong to more than one group. The *group_id* option is a number from 0 to 4294967295. The *ip_address* option specifies the IP address of the Call Agent.

mgcp command-queue

The **mgcp command-queue** command specifies the maximum number of MGCP commands that are queued while waiting for a response. The range of allowed values is from 1 to 4294967295. The default is 200. When the limit has been reached and a new command arrives, the command that has been in the queue for the longest time is removed.

mgcp gateway

The **mgcp gateway** command is used to specify which group of Call Agents are managing a particular gateway. The IP address of the gateway is specified with the *ip_address* option. The *group_id* option is a number from 0 to 4294967295 that must correspond with the *group_id* of the Call Agents that are managing the gateway. A gateway may only belong to one group.

clear mgcp and show mgcp

The **clear mgcp** command removes the MGCP configuration and resets the command queue limit to the default of 200.

The **show mgcp commands** command lists the number of MGCP commands in the command queue. The **show mgcp sessions** command lists the number of existing MGCP sessions. The **detail** option includes additional information about each command (or session) in the output.

Examples

The following example limits the MGCP command queue to 150 commands, allows Call Agents 10.10.11.5 and 10.10.11.6 to control gateway 10.10.10.115, and allows Call Agents 10.10.11.7 and 10.10.11.8 to control both gateways 10.10.10.116 and 10.10.10.117:

```
pixfirewall(config)# mgcp call-agent 10.10.11.5 101
pixfirewall(config)# mgcp call-agent 10.10.11.6 101
pixfirewall(config)# mgcp call-agent 10.10.11.7 102
pixfirewall(config)# mgcp call-agent 10.10.11.8 102
pixfirewall(config)# mgcp command-queue 150
pixfirewall(config)# mgcp gateway 10.10.10.115 101
pixfirewall(config)# mgcp gateway 10.10.10.116 102
pixfirewall(config)# mgcp gateway 10.10.10.117 102
```

The following are examples of the **show mgcp** command options:

```
pixfirewall# show mgcp commands
1 in use, 1 most used, 200 maximum allowed
CRCX, gateway IP: host-pc-2, transaction ID: 2052, idle: 0:00:07
pixfirewall# show mgcp commands detail
1 in use, 1 most used, 200 maximum allowed
CRCX, idle: 0:00:10
        Gateway IP      host-pc-2
        Transaction ID  2052
        Endpoint name   aaln/1
        Call ID         9876543210abcdef
        Connection ID
        Media IP        192.168.5.7
        Media port      6058

pixfirewall# show mgcp sessions
1 in use, 1 most used
Gateway IP host-pc-2, connection ID 6789af54c9, active 0:00:11

pixfirewall# show mgcp sessions detail
1 in use, 1 most used
Session active 0:00:14
        Gateway IP      host-pc-2
        Call ID         9876543210abcdef
        Connection ID   6789af54c9
        Endpoint name   aaln/1
        Media lcl port  6166
        Media rmt IP    192.168.5.7
        Media rmt port  6058
```

Related Commands

debug	Displays debug information for Media Gateway Control Protocol (MGCP) traffic.
fixup protocol	Enables the Media Gateway Control Protocol (MGCP) fixup. Use with the **mgcp** command to configure additional support for the MGCP fixup.
show conn	Displays all active connections. There is an MGCP **show conn** option and connection flag, “g”.
timeout	Sets the maximum idle time duration. (There is an MGCP timeout option.)

mroute

Configures a static multicast route.

[no] mroute *src smask in_if_name dst dmask out_if_name*

show mroute [*dst* [*src*]]

Syntax Description

dmask	The destination network address mask.
dst	The Class D address of the multicast group.
in_if_name	The input interface name to pass multicast traffic.
out_if_name	The output interface name to pass multicast traffic.

smask	The multicast source network address mask.
src	The IP address of the multicast source.

Command Modes

Configuration mode.

Usage Guidelines

The **mroute** command supports routing multicast traffic through the PIX Firewall.

The **show mroute** command displays the current multicast route table.

Examples

In the following example, the multicast sources are the inside interface and DMZ with no internal receivers:

```
multicast interface outside
multicast interface inside
multicast interface dmz

mroute 1.1.1.1 255.255.255.255 inside 230.1.1.2 255.255.255.255 outside
mroute 2.2.2.2 255.255.255.255 dmz 230.1.1.2 255.255.255.255 outside
```

mtu

Specify the maximum transmission unit (MTU) for an interface.

[**no**] **mtu** *if_name bytes*

show mtu

Syntax Description

bytes	The number of bytes in the MTU, in the range of 64 to 65,535 bytes. The value specified depends on the type of network connected to the interface.
if_name	The internal or external network interface name.

Command Modes

Configuration mode.

Usage Guidelines

The **mtu** command sets the size of data sent on a connection. Data larger than the maximum transmission unit (MTU) value is fragmented before being sent. The minimum value for *bytes* is 64 and the maximum is 65,535 bytes.

For PIX Firewall software Version 6.2, MTU size must be greater than or equal to 1500 for the Stateful Failover link and greater than or equal to 576 for the LAN-based failover link.

For PIX Firewall software Versions 5.2 through 6.1, MTU size must be greater than or equal to 256 bytes for the Stateful Failover link.

PIX Firewall supports the IP Path MTU Discovery mechanism, as defined in RFC 1191. IP Path MTU Discovery allows a host to dynamically discover and cope with differences in the maximum allowable maximum transmission unit (MTU) size of the various links along the path. Sometimes a PIX Firewall is unable to forward a datagram because it requires fragmentation (the packet is larger than the MTU you set for the interface), but the "don't fragment" (DF) bit is set. The network software sends a message to the sending host, alerting it to the problem. The host will have to fragment packets for the destination so that they fit the smallest packet size of all the links along the path.

For Ethernet interfaces, the default MTU is 1500 bytes in a block, which is also the maximum. This value is sufficient for most applications, but you can pick a lower number if network conditions warrant it.

The **no mtu** command resets the MTU block size to 1500 for Ethernet interfaces. The **show mtu** command displays the current block size. The **show interface** command also shows the MTU value.

Note For the MTU fragmentation to work properly when using L2TP, we recommend that the MTU size be set to 1380, in order to account for the L2TP header and IPSec header length.

Examples The following example shows the use of the **mtu** command with Ethernet:

```
interface ethernet1 auto
mtu inside 8192

show mtu
mtu outside 1500
mtu inside 8192
```

multicast

Enables multicast traffic to pass through the PIX Firewall. Includes an **igmp** subcommand mode for multicast support.

[**no**] **multicast interface** *interface_name*

clear multicast

show igmp [*group* | **interface** *interface_name*] [**detail**]

show multicast [**interface** *interface_name*]

Subcommands to the **multicast** command:

igmp forward interface *interface_name*

igmp access-group *id*

igmp version {**1** | **2**}

igmp join-group *group*

igmp max-groups *number*

igmp query-interval *seconds*

igmp query-max-response-time *seconds*

no igmp

clear igmp [*group* | **interface** *interface_name*]

Syntax Description

detail	Displays all information in the IGMP table.
id	Access control list ID.
group	The address of the multicast group.
igmp	Internet Group Management Protocol.
interface_name	The name of the interface on which to enable multicast traffic.
join-group	The multicast group to join.
max-groups	Specifies the maximum number of groups, from 0 to 2000. The default value is 500.
number	The maximum number of groups that can be joined.
query-interval	The query response time interval.
query-max-response-time	The maximum query response time interval.
seconds	Specifies the number of seconds to wait.

Command Modes

Configuration mode.

Usage Guidelines

The **multicast** command supports routing multicast traffic through the PIX Firewall.

The PIX Firewall **igmp** commands are subcommands of the **multicast** command.

The **clear igmp** [*group* | **interface** *interface_name*] command clears IGMP entries.

Note The PIX Firewall acts as an IGMP proxy but is not a multicast router.

The **show igmp** [*group* | **interface** *interface_name*] [**detail**] command displays the IGMP information for a multicast group, whether statically configured or dynamically created.

The **show multicast** [interface *interface_name*] command displays all or per-interface multicast settings. Also displays the IGMP configuration for any interface that is specified.

Examples

The following example shows use of the **multicast** command with corresponding **igmp** subcommands:

```
multicast interface outside
multicast interface inside
    igmp forward interface outside
    igmp join-group 224.1.1.1
```

The following is sample output from the **show igmp** command:

```
pixfirewall(config)# show igmp

  IGMP is enabled on interface inside
  Current IGMP version is 2
  IGMP query interval is 60 seconds
  IGMP querier timeout is 125 seconds
  IGMP max query response time is 10 seconds
  Last member query response interval is 1 seconds
  Inbound IGMP access group is
  IGMP activity: 0 joins, 0 leaves
  IGMP querying router is 10.1.3.1 (this system)

  IGMP Connected Group Membership
   Group Address    Interface            Uptime    Expires   Last Reported
```

name/names

Associate a name with an IP address.

[no] name *ip_address name*

[no] names

clear names

show names

Syntax Description

ip_address	The IP address of the host being named.
name	The name assigned to the IP address. Allowable characters are **a** to **z**, **A** to **Z**, **0** to **9**, a dash, and an underscore. The *name* cannot start with a number. If the name is over 63 characters long, the **name** command fails.

Command Modes

Configuration mode.

Usage Guidelines

Use the **name** command to identify a host by a text name. The names you define become like a host table local to the PIX Firewall. Because there is no connection to DNS or /etc/hosts on UNIX servers, use of this command is a mixed blessing—it makes configurations much more readable but introduces another level of abstraction to administer; not only do you have to add and delete IP addresses to your configuration as you do now, but with this command, you must ensure that the host names either match existing names or you have a map to list the differences.

The **name** command maps text strings to IP addresses. The **clear names** command clears the list of names from the PIX Firewall configuration. The **no names** command disables the use of the text names, but does not remove them from the configuration. The **show names** command lists the **name** command statements in the configuration.

Usage Notes

1. You must first use the **names** command before using the **name** command. Use the **name** command immediately after the **names** command and before you use the **write memory** command.
2. To disable displaying **name** values, use the **no names** command.
3. Only one name can be associated with an IP address.
4. Both the **name** and **names** command statements are saved in the configuration.
5. While the **name** command will let you assign a name to a network mask, no other PIX Firewall command requiring a mask will let you use the name as a mask value. For example, the following command is accepted.

```
name 255.255.255.0 class-C-mask
```

Note None of the commands in which a mask is required can process the "class-C-mask" as an accepted network mask.

Examples

In the example that follows, the **names** command enables use of the **name** command. The **name** command substitutes **pix_inside** for references to 192.168.42.3, and **pix_outside** for 209.165.201.3. The **ip address** commands use these names while assigning IP addresses to the network interfaces. The **no names** command disables the **name** command values from displaying. Subsequent use of the **names** command restores their display.

```
pixfirewall(config)# names
pixfirewall(config)# name 192.168.42.3 pix_inside
pixfirewall(config)# name 209.165.201.3 pix_outside
pixfirewall(config)# ip address inside pix_inside 255.255.255.0
pixfirewall(config)# ip address outside pix_outside 255.255.255.224

pixfirewall(config)# show ip address
System IP Addresses:
    inside ip address pix_inside mask 255.255.255.0
    outside ip address pix_outside mask 255.255.255.224

pixfirewall(config)# no names
pixfirewall(config)# show ip address
System IP Addresses:
    inside ip address 192.168.42.3 mask 255.255.255.0
    outside ip address 209.165.201.3 mask 255.255.255.224

pixfirewall(config)# names
pixfirewall(config)# show ip address
System IP Addresses:
    inside ip address pix_inside mask 255.255.255.0
    outside ip address pix_outside mask 255.255.255.224

pixfirewall(config)# show names
System IP Addresses:
    name 192.168.42.3 pix_inside
    name 209.165.201.3 pix_outside
```

nameif

Name interfaces and assign security level.

nameif {*hardware_id* | *vlan_id*} *if_name security_level*

clear nameif

show nameif

Syntax Description

hardware_id	The hardware name for the network interface that specifies the interface's slot location on the PIX Firewall motherboard. For more information on PIX Firewall hardware configuration, refer to the *Cisco PIX Firewall Hardware Installation Guide*. A logical choice for an Ethernet interface is **ethernet***n*. These names can also be abbreviated with any leading characters in the name, for example, **ether1** or **e2**.
if_name	A name for the internal or external network interface of up to 48 characters in length. By default, PIX Firewall names the inside interface "inside," the outside interface "outside," and any perimeter interface "intf*n*" where *n* is 2 through 5.
security_level	Enter **0** for the outside network or **100** for the inside network. Perimeter interfaces can use any number between **1** and **99**. By default, PIX Firewall sets the security level for the inside interface to **security100** and the outside interface to **security0**. The first perimeter interface is initially set to **security10**, the second to **security15**, the third to **security20**, and the fourth perimeter interface to **security25** (a total of 6 interfaces are permitted, with a total of 4 perimeter interfaces permitted). The word **security** in this command can also be abbreviated as **sec**, for example **sec10**. For access from a higher security to a lower security level, **nat** and **global** commands or **static** commands must be present. For access from a lower security level to a higher security level, **static** and **access-list** commands must be present. Interfaces with the same security level cannot communicate with each other. We recommend that every interface have a unique security level.
vlan_id	The VLAN identifier. For example: vlan10, vlan20, etc. (*vlan_id* is configured with the **interface** command.)

Command Modes

Configuration mode.

Usage Guidelines

The **nameif** command lets you assign a name to an interface. You can use this command to assign interface names if you have more than two network interface circuit boards in your PIX Firewall. The first two interfaces have the default names **inside** and **outside**. The **inside** interface has a default security level of 100, the **outside** interface has a default security level of 0. The **clear nameif** command reverts **nameif** command statements to default interface names and security levels.

Use **nameif** *hardware_id if_name security_level* to set name of a physical interface and use the **nameif** *vlan_id if_name security_level* command to set the name of a logical interface. Physical interfaces are one per each NIC, in place at boot time, and non-removable. Logical interfaces can be many-to-one for each NIC, are created at run time, and can be removed through software reconfiguration.

Usage Notes

1. If you change the *hardware_id* of the outside interface; for example, from ethernet0 to ethernet1, PIX Firewall changes every reference to the outside interface in your configuration to inside, which can cause problems with **route**, **ip**, and other command statements that affect the flow of traffic through the PIX Firewall.
2. After changing a **nameif** command, use the **clear xlate** command.
3. The inside interface cannot be renamed or given a different security level. The outside interface can be renamed, but not given a different security level.
4. An interface is always "external" with respect to another interface that has a higher security level.

Examples

The following example shows how to use the **nameif** *hardware_id if_name security_level* command:

```
nameif ethernet2 perimeter1 sec50
nameif ethernet3 perimeter2 sec20

The following example shows how to use the nameif vlan_id if_name security_level command:
nameif vlan10 perimeter3 sec10

The following example is a configuration that uses both physical and VLAN interfaces:
nameif ethernet0 outside security0
nameif ethernet1 intf6 security90
nameif ethernet2 dmz security50
nameif vlan4 intf4 security10
nameif vlan5 intf5 security10
nameif vlan10 intf5 security10
```

Related Commands

interface	Sets network interface parameters and configures VLANs.

nat

Associate a network with a pool of global IP addresses.

[**no**] **nat** [(*local_interface*)] *id local_ip* [*mask* [**dns**] [**outside** | [**norandomseq**] [*max_conns* [*emb_limit*]]]]

[**no**] **nat** [(*local_interface*)] *id* **access-list** *acl_name* [**dns**] [**outside** | [**norandomseq**] [*max_conns* [*emb_limit*]]]

[**no**] **nat** [(*local_interface*)] **0 access-list** *acl_name* [**outside**]

clear nat

show nat

Syntax Description

access-list	Lets you identify local traffic for network address translation (NAT) by specifying the local and destination addresses (or ports). This feature is known as policy NAT. **Note** Even though NAT exemption (**nat 0 access-list)** uses an access list, this function is not the same as policy NAT. NAT exemption does not use ports in the access list. You can only include **permit** statements in the access list. Local traffic is matched to the first matching policy NAT statement. See the "Order of NAT Commands Used to Match Local Addresses" section on page 7-17 for more information.
acl_id	Specifies the access list name.
clear nat	Removes **nat** command statements from the configuration.
dns	Specifies to use the created translation to rewrite the DNS address record.
emb_limit	Specifies the maximum number of embryonic connections per host. An embryonic connection is a connection request that has not finished the necessary handshake between source and destination. Set a small value for slower systems, and a higher value for faster systems. The default is 0, which means unlimited embryonic connections. The embryonic connection limit lets you prevent a type of attack where processes are started without being completed. When the embryonic limit has been surpassed, the TCP intercept feature intercepts TCP synchronization (SYN) packets from clients to servers on a higher security level. 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 combines the two half-connections together transparently. Thus, connection attempts from unreachable hosts will never reach the server. The PIX firewall accomplishes TCP intercept functionality using SYN cookies. **Note** This option does not apply to outside NAT. The TCP intercept feature only applies to hosts or servers on a higher security level. If you set the *emb_limit* as well as the **outside** option, the *emb_limit* is ignored.
(*local_interface*)	Specifies the name of the network interface, as defined by the **nameif** command, through which the hosts or network designated by *local_ip* or **access-list** *acl_id* are accessed. You must enter the interface name in parentheses. If you do not enter the interface name, then the default is **inside**.
local_ip	Specifies the addresses to translate. You can use **0.0.0.0** (or **0** for short) to identify all hosts. Local traffic is matched to a **nat** statement using the best match. See the "Order of NAT Commands Used to Match Local Addresses" section on page 7-17 for more information.
mask	Specifies the IP netmask to apply to *local_ip*. If you do not specify a mask, the PIX Firewall derives the network mask from the class of the IP address. For example, the command **nat 0 10.130.36.0** causes all addresses in the 10.0.0.0 network to be translated and not only those in the 10.130.36.0 network. For this reason, you should specify the network mask when configuring an IP address that is not classful. You must also specify the mask to set other options, such as **outside**.

max_conns	Specifies the maximum number of simultaneous TCP and UDP connections for the entire subnet. The default is 0, which means unlimited connections. (Idle connections are closed after the idle timeout specified by the **timeout conn** command.) **Note** This option does not apply to outside NAT. The firewall only tracks connections from a higher security interface to a lower security interface. If you set *max_conns* as well as the **outside** option, the *max_conns* option is ignored.
nat_id	Specifies an integer for the NAT ID. For regular NAT, this integer is between 1 and 2147483647. For policy NAT (**nat** *id* **access-list**), this integer is between 1 and 65535. Identity NAT (**nat 0**) and NAT exemption (**nat 0 access-list**) use the NAT ID of **0**. See the "nat 0 (Identity NAT)" section on page 7-16 and the "nat 0 access-list (NAT Exemption)" section on page 7-16 for more information about NAT identity and exemption.
norandomseq	Disables TCP Initial Sequence Number (ISN) randomization protection. Only use this option if another inline firewall is also randomizing sequence numbers and the result is scrambling the data. Without this protection, inside hosts with weak self-ISN protection become more vulnerable to TCP connection hijacking. **Note** This option does not apply to outside NAT. The firewall only randomizes the ISN that is generated by the host/server on the higher security interface. If you set **norandomseq** as well as the **outside** option, the **norandomseq** option is ignored.
outside	If this interface is on a lower security level than the interface you identify by the matching **global** statement, then you must enter **outside**. This feature is called outside NAT or bidirectional NAT. **Note** Starting with PIX Firewall 6.3.2, source translation is performed before destination translation. For this reason, if the source NAT policy allows the connection, the xlate will be created, even if the traffic is denied by the destination policy.

Command Modes

Configuration mode.

Usage Guidelines

Network Address Translation (NAT) substitutes the local address of a packet with a global address that is routable on the destination network.

When hosts on a higher security interface (inside) access hosts on a lower security interface (outside), you must configure NAT on the inside hosts *or* specifically configure the inside interface to bypass NAT.

An inside host can communicate with the untranslated local address of the outside host without any special configuration on the outside interface. However, you can also optionally perform NAT on the outside network.

The **nat** command identifies the local addresses for translation using dynamic NAT or port address translation (PAT). The **global** command identifies the global addresses used for translation on a given destination interface. Each **nat** statement matches a **global** statement by comparing the NAT ID on each

statement. If you bypass NAT using identity NAT or NAT exemption, then no **global** command is required. See the "nat 0 (Identity NAT)" section on page 7-16 and the "nat 0 access-list (NAT Exemption)" section on page 7-16 for more information on bypassing NAT.

After changing or removing a **nat** command statement, use the **clear xlate** command.

You can use the **no nat** command to remove a **nat** command statement.

Note The firewall does not support NAT for a Call Manager (CM) inside the firewall with IP phones outside the firewall (that need to register with it). This is because when the IP phone needs to register with the CM it does so through TFTP, but the firewall does not NAT TFTP messages.

The PIX Firewall does not support outside NAT for non-H.323 multimedia applications or between overlapping network addresses.

Dynamic NAT and PAT

Dynamic NAT translates a group of local addresses to a pool of global addresses that are routable on the destination network. The global pool can include fewer addresses than the local group. When a local host accesses the destination network, the FWSM assigns it an IP address from the global pool. Because the translation is only in place for the duration of the connection, a given user does not keep the same IP address between connections. Users on the destination network, therefore, cannot reliably initiate a connection to a host that uses dynamic NAT (even if the connection is allowed by an access list). Not only can you not predict the IP address of the host, but the host does not have a global address unless the host is the initiator. See the **static** command for reliable access to hosts.

PAT translates a group of local addresses to a single global IP address combined with a unique source port (above 1024). When a local host accesses the destination network, the FWSM assigns it the global IP address and then a unique port number. Each host receives the same IP address, but because the source port numbers are unique, the responding traffic, which includes the IP address and port number as the destination, can be assigned to the correct host. Because there are over 64,000 ports available, you are unlikely to run out of addresses, which can happen with dynamic NAT.

Like dynamic NAT, the translation is only in place for the duration of the connection, so a given user does not keep the same port number between connections.

PAT allows you to use a single global address, thus conserving routable addresses. You can even use the destination interface IP address as the PAT address. PAT does not work with multimedia applications that have an inbound data stream different from the outgoing control path.

Dynamic NAT has these disadvantages:

- If the global pool has fewer addresses than the local group, you could run out of addresses if the traffic is more than expected.

  Use PAT if this event occurs often.

- You have to use a large number of routable addresses in the global pool; if the destination network requires registered addresses, such as the Internet, you might encounter a shortage of usable addresses.

The advantage of dynamic NAT is that some protocols cannot use PAT, which does not work with applications that have an inbound data stream on one port and the outgoing control path on another, such as multimedia applications.

nat Vs. static Commands

The rule of thumb is that for access from a higher security level interface to a lower security level interface, use the **nat** command. From lower security level interface to a higher security level interface, use the **static** command.

Table 7-1 helps you decide when to use the **nat** or **static** commands for access between the various interfaces in the PIX Firewall. For this table, assume that the security levels are 40 for dmz1 and 60 for dmz2.

Table 7-1 Interface Access Commands by Interface

From This Interface	To This Interface	Use This Command
inside	outside	**nat**
inside	dmz1	**nat**
inside	dmz2	**nat**
dmz1	outside	**nat**
dmz1	dmz2	**static**
dmz1	inside	**static**
dmz2	outside	**nat**
dmz2	dmz1	**nat**
dmz2	inside	**static**
outside	dmz1	**static**
outside	dmz2	**static**
outside	inside	**static**

nat 0 (Identity NAT)

The **nat 0** command enables identity NAT. Use this command to bypass NAT and allow the local addresses to be used unchanged. Adaptive Security remains in effect with the **nat 0** command. Both the **nat 0** command and the **nat 0 access-list** command (NAT exemption) may be configured concurrently in PIX Firewall software Version 5.3 and higher.

It is important to understand the difference between identity NAT and NAT exemption. With identity NAT, you can accept the inbound traffic only when the traffic is initiated from the inside and after the xlate is created. NAT exemption allows traffic whenever it matches the referenced ACL, regardless of whether or not there is already an xlate. Identity NAT allows you to set additional NAT parameters, such as **norandomseq**. NAT exemption allows only the **outside** option.

The **nat 0 10.2.3.0** command means let those IP addresses in the 10.2.3.0 net appear on the outside without translation. All other hosts are translated depending on how their **nat** or **static** command statements appear in the configuration.

nat 0 access-list (NAT Exemption)

The **nat 0 access-list** command disables NAT, specifically proxy ARPing, for the IP addresses specified by the ACL referenced by *acl_id*. (The *acl_id* is the name you use to identify the **access-list** command statement.) This feature is known as NAT exemption. NAT exemption is not backward compatible with PIX Firewall software Version 5.2 or earlier versions.

This feature is useful in a Virtual Private Network (VPN) configuration where traffic between private networks should be exempted from NAT.

While NAT exemption lets you exempt traffic that is matched by the **access-list** command statement from NAT services, Adaptive Security remains in effect. The extent to which the inside hosts are accessible from the outside depends on the **access-list** command statements that permit inbound access; NAT exemption allows both inbound and outbound traffic no matter which side initiates, as long as it is permitted by the referenced ACL.

The ACL must have only **permit** statements. Unlike policy NAT, the PIX Firewall ignores any port setting in your ACL command statement and so NAT exemption cannot be used to permit or deny traffic on a per-port basis.

nat outside (Outside NAT)

The **nat outside** option lets you enable or disable outside NAT, which translates the source address of a connection coming from a lower security interface to higher interface. This feature is also called bidirectional NAT.

If you enable outside dynamic NAT on an interface, then you must configure explicit NAT policy for all hosts on the interface that need to initiate connections to inside networks. If you want to translate some hosts, but not others, then use identity NAT or NAT exemption (**nat 0** or **nat 0 access-list**) to disable address translation for these additional hosts.

Do not specify the **norandomseq** or *emb_limit* options for outside NAT. These options only apply to traffic initiated from a higher security interface.

Note Enabling outside PAT can make the firewall more susceptible to flood DoS attack. To mitigate this, we recommend that the address range selected with the **nat** *nat_id local_ip mask* **outside** command be as restrictive as possible. In addition, the connection limit should be set to a value that takes into consideration the memory capacity of the firewall. In general, a PAT session is composed of a PAT xlate and a UDP or TCP connection. A PAT xlate consumes about 120 bytes and a TCP or UDP connection consumes about 250 bytes.

nat *nat_id* access-list (Policy NAT)

When you use an access list with the **nat** command for any NAT ID other than 0, then you enable policy NAT.

Policy NAT lets you identify local traffic for address translation by specifying the source and destination addresses (or ports) in an access list. Regular NAT uses source addresses/ports only, whereas policy NAT uses both source and destination addresses/ports.

Note All types of NAT support policy NAT except for NAT exemption (**nat 0 access-list**). NAT exemption uses an ACL to identify the local addresses, but differs from policy NAT in that the ports are not considered.

With policy NAT, you can create multiple NAT or static statements that identify the same local address as long as the source/port and destination/port combination is unique for each statement. You can then match different global addresses to each source/port and destination/port pair.

Order of NAT Commands Used to Match Local Addresses

The firewall matches local traffic to NAT commands in the following order:

1. **nat 0 access-list** (NAT exemption)—In order, until the first match. For example, you could have overlapping local/destination addresses in multiple **nat** commands, but only the first command is matched.

2. **static** (static NAT)—In order, until the first match. Because you cannot use the same local address in static NAT or static PAT commands, the order of **static** commands does not matter. Similarly, for static policy NAT, you cannot use the same local/destination address and port across multiple statements.
3. **static** {**tcp** | **udp**} (static PAT)—In order, until the first match. Because you cannot use the same local address in static NAT or static PAT commands, the order of **static** commands does not matter. Similarly, for static policy NAT, you cannot use the same local/destination address and port across multiple statements.
4. **nat** *nat_id* **access-list** (policy NAT)—In order, until the first match. For example, you could have overlapping local/destination ports and addresses in multiple **nat** commands, but only the first command is matched.
5. **nat** (regular NAT)—Best match. The order of the NAT commands does not matter. The **nat** statement that best matches the local traffic is used. For example, you can create a general statement to translate all addresses (0.0.0.0) on an interface. If you also create a statement to translate only 10.1.1.1, when 10.1.1.1 makes a connection, the specific statement for 10.1.1.1 is used because it matches the local traffic best.

If you configure multiple **global** statements on the same NAT ID, the **global** statements are used in this order:

1. No **global** if using **nat 0** (identity NAT).
2. Dynamic NAT **global**.
3. PAT **global**.

Examples

The **nat 0** (identity NAT) command allows traffic to be initiated from the local host only.

If you want the addresses to be visible from the outside network, use NAT exemption, or use the **static** command as follows:

```
nat (inside) 0 209.165.201.0 255.255.255.224
static (inside, outside) 209.165.201.0 209.165.201.0 netmask 255.255.255.224
access-list acl_out permit host 10.0.0.1 209.165.201.0 255.255.255.224 eq ftp
access-group acl_out in interface outside

nat (inside) 0 209.165.202.128 255.255.255.224
static (inside, outside) 209.165.202.128 209.165.202.128 netmask 255.255.255.255
access-list acl_out permit tcp host 10.0.0.1 209.165.202.128 255.255.255.224 eq ftp
access-group acl_out in interface outside
```

The following example shows use of the **nat 0 access-list** command (NAT exemption) to permit internal host 10.1.1.15, which is accessible through the inside interface, to bypass NAT when connecting to outside host 10.2.1.3.

```
access-list no-nat permit ip host 10.1.1.15 host 10.2.1.3
nat (inside) 0 access-list no-nat
```

The following commands use NAT exemption on a PIX Firewall with three interfaces:

```
access-list all-ip-packet permit ip 0 0 0 0
nat (dmz) 0 access-list all-ip-packet
nat (inside) 0 access-list all-ip-packet
```

Given outbound traffic and the following example, for the **nat** command statements with a *nat_id* of **1**, any of the hosts on the 10.1.1.0 network are translated to the range of 209.165.201.25-209.165.201.27. After all three addresses have been used, the translation rule starts using 209.165.201.30 as the PAT address. For the **nat** command statements with a *nat_id* of **3**, all of the hosts on the 10.1.3.0 network are translated to the outside IP address of the FWSM using PAT.

```
nat (inside) 1 10.1.1.0 255.255.255.0
global (outside) 1 209.165.201.25-209.165.201.27 netmask 255.255.255.224
global (outside) 1 209.165.201.30

nat (inside) 3 10.1.3.0 255.255.255.0
global (outside) 3 209.165.201.30
```

The following example specifies with **nat** command statements that all the hosts on the 10.0.0.0 and 10.3.3.0 inside networks can start outbound connections. The **global** command statements create unique pools of global addresses for those hosts that cannot overlap.

```
nat (inside) 1 10.0.0.0 255.0.0.0
global (outside) 1 209.165.201.24-209.165.201.27 netmask 255.255.255.224
global (outside) 1 209.165.201.30

nat (inside) 3 10.3.3.0 255.255.255.0
global (outside) 3 209.165.201.10-209.165.201.23 netmask 255.255.255.224
```

The following policy NAT example shows a host on the 10.1.2.0/24 network accessing two different servers. When the host accesses the server at 209.165.201.11, the local address is translated to 209.165.202.129. When the host accesses the server at 209.165.200.225, the local address is translated to 209.165.202.130.

```
access-list NET1 permit ip 10.1.2.0 255.255.255.0 209.165.201.0 255.255.255.224
access-list NET2 permit ip 10.1.2.0 255.255.255.0 209.165.200.224 255.255.255.224
nat (inside) 1 access-list NET1
global (outside) 1 209.165.202.129 255.255.255.255
nat (inside) 2 access-list NET2
global (outside) 2 209.165.202.130 255.255.255.255
```

The following policy NAT example shows the use of source and destination ports. The host on the 10.1.2.0/24 network accesses a single host for both web services and Telnet services. When the host accesses the server for web services, the local address is translated to 209.165.202.129. When the host accesses the same server for Telnet services, the local address is translated to 209.165.202.130.

```
access-list WEB permit tcp 10.1.2.0 255.255.255.0 209.165.201.11 255.255.255.255 eq 80
access-list TELNET permit tcp 10.1.2.0 255.255.255.0 209.165.201.11 255.255.255.255 eq 23
nat (inside) 1 access-list WEB
global (outside) 1 209.165.202.129 255.255.255.255
nat (inside) 2 access-list TELNET
global (outside) 2 209.165.202.130 255.255.255.255
```

Related Commands

access-list	Configures access control.
global	Configures global address pools, or designates a PAT (Port Address Translation) address.
interface	Sets network interface parameters and configures VLANs.
nameif	Assigns a name to an interface.
static	Configures a one-to-one address translation rule.

ntp

Synchronizes the PIX Firewall with a network time server using the Network Time Protocol (NTP).

[**no**] **ntp authenticate**

[**no**] **ntp authentication-key** *number* **md5** *value*

ntp server *ip_address* [**key** *number*] **source** *if_name* [**prefer**]

no ntp server *ip_address*

[**no**] **ntp trusted-key** *number*

clear ntp

show ntp

show ntp associations [**detail**]

show ntp status

Syntax Description

associations	The network time server associations.
authenticate	Enables NTP authentication. If enabled, the PIX Firewall requires authentication before synchronizing with an NTP server.
authentication-key	Defines the authentication keys for use with other NTP commands.
detail	Provides additional detail on the network time servers.
if_name	Specifies the interface to use to send packets to the network time server.
ip_address	The IP address of the network time server with which to synchronize.
key	Specifies the authentication key.
md5	The encryption algorithm.
number	The authentication key number (1 to 4294967295).
prefer	Designates the network time server specified as the preferred server with which to synchronize time.
server	The network time server.
source	Specifies the network time source.
status	Displays NTP clock information.
trusted-key	Specifies the trusted key against which to authenticate.
value	The key value, an arbitrary string of up to 32 characters. The key value is displayed as "**********" when the configuration is viewed by the **write terminal** or **show tech-support** commands.

Command Modes

Configuration mode.

Usage Guidelines

The **ntp** command synchronizes the PIX Firewall with the network time server that is specified and authenticates according to the authentication options that are set.

The authentication keys for the **ntp** commands are defined in the **ntp authentication-key** command. If authentication is used, the PIX Firewall and NTP server must be configured with the same key.

If authentication is enabled, use the **ntp trusted-key** command to define one or more key numbers that the NTP server needs to provide in its NTP packets for the PIX Firewall to accept synchronization with the NTP server.

The PIX Firewall listens for NTP packets (port 123) only on interfaces that have an NTP server configured through the **ntp server** command. NTP packets that are not responses from a request by the PIX Firewall are dropped.

The **ntp authenticate** command enables NTP authentication.

The **clear ntp** command removes the NTP configuration, including disabling authentication and removing all authentication keys and NTP server designations.

show ntp commands

To view information about the NTP configuration and status, use the **show ntp, show ntp associations** [**detail**], or **show ntp status** commands.

The **show ntp** command displays the current NTP configuration.

The **show ntp associations** [**detail**] command displays the configured network time server associations.

The **show ntp status** command displays the NTP clock information.

The following is sample output from the **show ntp associations** command:

```
pixfirewall> show ntp associations
     address         ref clock     st  when  poll  reach  delay  offset    disp
 ~172.31.32.2      172.31.32.1       5    29  1024  377     4.2   -8.59     1.6
+~192.168.13.33    192.168.1.111     3    69   128  377     4.1    3.48     2.3
*~192.168.13.57    192.168.1.111     3    32   128  377     7.9   11.18     3.6
* master (synced), # master (unsynced), + selected, - candidate, ~ configured
```

Table 7-2 describes the values in the **show ntp associations** command output:

Table 7-2 Output Description from ntp association Command

Output	Description
*	Synchronized to this peer
#	Almost synchronized to this peer
+	Peer selected for possible synchronization
-	Peer is a candidate for selection
~	Peer is statically configured
address	Address of peer.
ref clock	Address of reference clock of peer.
st	Stratum of peer.
when	Time since last NTP packet was received from peer.
poll	Polling interval (in seconds).
reach	Peer reachability (bit string, in octal).
delay	Round-trip delay to peer (in milliseconds).

Table 7-2 Output Description from ntp association Command

Output	Description
offset	Relative time of peer clock to local clock (in milliseconds).
disp	Dispersion.

The following is sample output from the **show ntp association detail** command:

```
pixfirewall(config)# show ntp associations detail
172.23.56.249 configured, our_master, sane, valid, stratum 4
ref ID 172.23.56.225, time c0212639.2ecfc9e0 (20:19:05.182 UTC Fri Feb 22
2002)
our mode client, peer mode server, our poll intvl 128, peer poll intvl 128
root delay 38.04 msec, root disp 9.55, reach 177, sync dist 156.021
delay 4.47 msec, offset -0.2403 msec, dispersion 125.21
precision 2**19, version 3
org time c02128a9.731f127b (20:29:29.449 UTC Fri Feb 22 2002)
rcv time c02128a9.73c1954b (20:29:29.452 UTC Fri Feb 22 2002)
xmt time c02128a9.6b3f729e (20:29:29.418 UTC Fri Feb 22 2002)
filtdelay =     4.47    4.58    4.97    5.63    4.79    5.52    5.87
0.00
filtoffset =   -0.24   -0.36   -0.37    0.30   -0.17    0.57   -0.74
0.00
filterror =     0.02    0.99    1.71    2.69    3.66    4.64    5.62
16000.0
```

Table 7-3 describes the values in the **show ntp association detail** command output:

Table 7-3 Output Description from ntp association detail Command

Output	Description
configured	Peer was statically configured.
dynamic	Peer was dynamically discovered.
our_master	Local machine is synchronized to this peer.
selected	Peer is selected for possible synchronization.
candidate	Peer is a candidate for selection.
sane	Peer passes basic sanity checks.
insane	Peer fails basic sanity checks.
valid	Peer time is believed to be valid.
invalid	Peer time is believed to be invalid.
leap_add	Peer is signalling that a leap second will be added.
leap-sub	Peer is signalling that a leap second will be subtracted.
unsynced	Peer is not synchronized to any other machine.
ref ID	Address of machine peer is synchronized to.
time	Last time stamp peer received from its master.
our mode	Our mode relative to peer (active/passive/client/server/bdcast/bdcast client).
peer mode	Peer's mode relative to us.
our poll intvl	Our poll interval to peer.

Table 7-3 Output Description from ntp association detail Command (continued)

Output	Description
peer poll intvl	Peer's poll interval to us.
root delay	Delay along path to root (ultimate stratum 1 time source).
root disp	Dispersion of path to root.
reach	Peer reachability (bit string in octal).
sync dist	Peer synchronization distance.
delay	Round-trip delay to peer.
offset	Offset of peer clock relative to our clock.
dispersion	Dispersion of peer clock.
precision	Precision of peer clock in hertz.
version	NTP version number that peer is using.
org time	Originate time stamp.
rcv time	Receive time stamp.
xmt time	Transmit time stamp.
filtdelay	Round-trip delay (in milliseconds) of each sample.
filtoffset	Clock offset (in milliseconds) of each sample.
filterror	Approximate error of each sample.

The following is sample output from the **show ntp status** command:

```
pixfirewall(config)# show ntp status
Clock is synchronized, stratum 5, reference is 172.23.56.249
nominal freq is 99.9984 Hz, actual freq is 100.0266 Hz, precision is 2**6
reference time is c02128a9.73c1954b (20:29:29.452 UTC Fri Feb 22 2002)
clock offset is -0.2403 msec, root delay is 42.51 msec
root dispersion is 135.01 msec, peer dispersion is 125.21 msec
```

Table 7-4 describes the values in the **show ntp status** command output:

Table 7-4 Output Description from ntp status Command

Output	Description
synchronized	System is synchronized to an NTP peer.
unsynchronized	System is not synchronized to any NTP peer.
stratum	NTP stratum of this system.
reference	Address of peer to which the system is synchronized.
nominal freq	Nominal frequency of system hardware clock.
actual freq	Measured frequency of system hardware clock.
precision	Precision of the clock of this system (in hertz).
reference time	Reference time stamp.
clock offset	Offset of the system clock to synchronized peer.
root delay	Total delay along path to root clock.

Table 7-4 Output Description from ntp status Command (continued)

Output	Description
root dispersion	Dispersion of root path.
peer dispersion	Dispersion of synchronized peer.

Examples

The following is sample output from the **show ntp** command:

```
pixfirewall(config)# show ntp
ntp authentication-key 1234 md5 ********
ntp authenticate
ntp trusted-key 1234
ntp server 10.10.1.2 key 1234 source inside prefer
pixfirewall(config)#
```

The following is sample output from the **show ntp associations** command:

```
pixfirewall(config)# show ntp associations
  address           ref clock         st  when  poll reach   delay  offset  disp
 *~172.23.56.249    172.23.56.225     4   113   128  177     4.5   -0.24   125.2
 * master (synced), # master (unsynced), + selected, - candidate, ~ configured
```

The following is sample output from the **show ntp associations detail** command:

```
pixfirewall(config)# show ntp associations detail
172.23.56.249 configured, our_master, sane, valid, stratum 4
ref ID 172.23.56.225, time c0212639.2ecfc9e0 (20:19:05.182 UTC Fri Feb 22 2002)
our mode client, peer mode server, our poll intvl 128, peer poll intvl 128
root delay 38.04 msec, root disp 9.55, reach 177, sync dist 156.021
delay 4.47 msec, offset -0.2403 msec, dispersion 125.21
precision 2**19, version 3
org time c02128a9.731f127b (20:29:29.449 UTC Fri Feb 22 2002)
rcv time c02128a9.73c1954b (20:29:29.452 UTC Fri Feb 22 2002)
xmt time c02128a9.6b3f729e (20:29:29.418 UTC Fri Feb 22 2002)
filtdelay =     4.47    4.58    4.97    5.63    4.79    5.52    5.87    0.00
filtoffset =   -0.24   -0.36   -0.37    0.30   -0.17    0.57   -0.74    0.00
filterror =     0.02    0.99    1.71    2.69    3.66    4.64    5.62    16000.0
```

The following is sample output from the **show ntp status** command:

```
pixfirewall(config)# show ntp status
Clock is synchronized, stratum 5, reference is 172.23.56.249
nominal freq is 99.9984 Hz, actual freq is 100.0266 Hz, precision is 2**6
reference time is c02128a9.73c1954b (20:29:29.452 UTC Fri Feb 22 2002)
clock offset is -0.2403 msec, root delay is 42.51 msec
root dispersion is 135.01 msec, peer dispersion is 125.21 msec
```

Related Commands

clock	Sets the date and time of firewall.

object-group

Defines object groups that you can use to optimize your configuration. Objects such as hosts, protocols, or services can be grouped, and then you can issue a single command using the group name to apply to every item in the group.

[**no**] **object-group icmp-type** *grp_id*

ICMP type group subcommands:
description *description_text*
icmp-object *icmp_type*
group-object *grp_id*

[**no**] **object-group network** *grp_id*

network group subcommands:
description *description_text*
network-object host *host_addr*
network-object *host_addr mask*
group-object *grp_id*

[**no**] **object-group protocol** *grp_id*

protocol group subcommands:
description *description_text*
protocol-object *protocol*
group-object *grp_id*

[**no**] **object-group service** *grp_id* {**tcp | udp | tcp-udp**}

service group subcommands:
description *description_text*
port-object range *begin_service end_service*
port-object eq *service*
group-object *grp_id*

clear object-group [*grp_type]*

show object-group [**id** *grp_id* | *grp_type*]

Note Enter **no** in front of a subcommand to remove the configuration within an object group.

Syntax Description.

begin_service	Used with the **range** keyword, the decimal number or name of a TCP or UDP port that is the beginning value for a range of services.
description *description_text*	A subcommand of the **object-group** command that enables users to add a description of up to 200 characters to an object-group. The starting position of the description text is the character right after the whitespace (a blank or a tab) following the **description** keyword.
end_service	Used with the **range** keyword, the decimal number or name of a TCP or UDP port that is the ending value for a range of services.

eq *service*	Specifies the decimal number or name of a TCP or UDP port for a particular service object.
group-object	The **group-object** subcommand is used to add a group of objects that are themselves members of another object group.
grp_id	Required parameter that identifies the object group (one to 64 characters). Can be any combination of letters, digits, and the "_", "-", "." characters.
grp_type	*The type of group, either ICMP type, network, protocol, or service.*
host	Keyword used with the *host_addr* parameter to define a host object.
host_addr	The host IP address or host name (if the host name is already defined using the **name** command).
icmp-object	The **object-group icmp-type** subcommand used to add ICMP objects to an ICMP-type object group.
icmp-type	Defines a group of ICMP types such as echo and echo-reply. After entering the main **object-group icmp-type** command, add ICMP objects to the ICMP type group with the **icmp-object** and the **group-object** subcommand.
icmp_type	The decimal number or name of an ICMP type.
mask	The netmask. Used with *net_addr* to define a subnet object.
net_addr	The network address. Used with *netmask* to define a subnet object.
network	Defines a group of hosts or subnet IP addresses. After entering the main **object-group network** command, add network objects to the network group with the **network-object** and the **group-object** subcommand.
network-object	The **object-group network** subcommand used to add network objects to a network object group.
obj_grp_id	The name of a previously defined object group. For object groups to be grouped together, they must be of the same type. For example, you can group two or more network object groups together, but you cannot group a protocol group and a network group together.
object-group	The main object grouping command. The keyword after it specifies the type of object group that is being defined. After entering this main command with the type indicator keyword, you are in subcommand mode where you explicitly define individual group members using the **object-group** subcommands.
port-object	The **object-group service** subcommand used to add port objects to a service object group.
protocol	Defines a group of protocols such as TCP and UDP. After entering the main **object-group protocol** command, add protocol objects to the protocol group with the **protocol-object** and the **group-object** subcommand.
protocol	The protocol name or number. (For example, UDP is 17 and TCP is 6.)
protocol-object	The **object-group protocol** subcommand used to add protocol objects to a protocol object group.
range	Keyword indicating that the range parameters follow.
service	Defines a group of TCP/UDP port specifications such as "eq smtp" and "range 2000 2010." After entering the main **object-group service** command, add port objects to the service group with the **port-object** and the **group-object** subcommand.
tcp	Specifies that service group is used for TCP.

tcp-udp	Specifies that service group can be used for TCP and UDP.
udp	Specifies that service group is used for UDP.

Command Modes

Configuration mode.

Usage Guidelines

When a group is defined with the **object-group** command and then used in a PIX Firewall command, the command applies to every item in that group. This can significantly reduce your configuration size.

Once an object group is defined, the keyword **object-group** must be used before the group name in all applicable PIX Firewall commands, for example:

show object-group *group_name*

where *group_name* is the name of the group.

The following are two examples of the use of an object group once it is defined:

```
conduit permit tcp object-group group_name any
access-list acl_id permit tcp any object-group group_name
```

Additionally, the **access-list** and **conduit** command parameters can be grouped as follows in Table 7-5.

Table 7-5 Object Groups to Replace Individual Parameters

Instead of using individual parameters...	...use the following object group:
protocol	**object-group** *protocol*
host and subnet	**object-group** *network*
service	**object-group** *service*
icmp_type	**object-group** *icmp_type*

You can group commands hierarchically; an object group can be a member of another object group.

To use object groups, you must do the following:

- The keyword **object-group** must be used before the object group name in all commands.

  For example:

  ```
  access-list acl permit tcp object-group remotes object-group locals object-group
  eng_svc
  ```

  where *remotes* and *locals* are sample object group names.
- The object group must be non-empty.
- An object group cannot be removed or emptied if it is currently being used in a command.

After a main **object-group** command is entered, the command mode changes to its corresponding subcommand mode. The object group is then defined in the subcommand mode. The active mode is indicated in the command prompt format. For example, the prompt in the configuration terminal mode appears as follows:

```
pix_name (config)#
```

where *pix_name* is the name of the PIX Firewall.

However, when the **object-group** command is entered, the prompt appears as follows:

```
pix_name (config-type)#
```

where *pix_name* is the name of the PIX Firewall and *type is the object-group type.*

Use **exit**, **quit**, or any valid config-mode command such as the **access-list** command to close an **object-group** subcommand mode and exit the **object-group** main command.

object groupingUse the **no object-group** command form to remove a group of previously defined **object-group** commands. The **clear object-group** command form can also be used.

The **show object-group** command displays all defined object groups by their *grp_id* when the **show object-group id** *grp_id* command form is entered, and by their group type when the **show object-group** *grp_type* command form is entered. When you enter the **show object-group** command without a parameter, all defined object groups are shown.

When entered without a parameter, the **clear object-group** command removes all defined object groups that are not being used in a command. Using *grp_type* parameter removes all defined object groups that that are not being used in a command for that group type only.

For use in the **object-group icmp-type** command, Table 7-6 lists ICMP type numbers and names:

Table 7-6 object groupingICMP Types

Number	Name of ICMP Type
0	echo-reply
3	unreachable
4	source-quench
5	redirect
6	alternate-address
8	echo
9	router-advertisement
10	router-solicitation
11	time-exceeded
12	parameter-problem
13	timestamp-request
14	timestamp-reply
15	information-request
16	information-reply
17	mask-request
18	mask-reply
31	conversion-error
32	mobile-redirect

Usage Notes

1. You can use all other PIX Firewall commands in subcommand mode, including the **show** and **clear** commands.
2. Subcommands appear indented when displayed or saved by the **show config**, **write**, or **config** commands.

3. Subcommands have the same command privilege level as the main command.
4. When more than one object group is used in an **access-list** or **conduit** command, the elements of all object groups used in the command are cross-concatenated together, starting with the first group's elements concatenated the second group's elements, then the first and second group's elements concatentated together with the third group's elements, and so on.

Examples

The following example shows how to use the **object-group icmp-type** subcommand mode to create a new icmp-type object group:

```
pixfirewall(config)# object-group icmp-type icmp-allowed
    pixfirewall(config-icmp-type)#icmp-object echo
    pixfirewall(config-icmp-type)#icmp-object time-exceeded
    pixfirewall(config-icmp-type)#exit
```

The following example shows how to use the **object-group network** subcommand to create a new network object group:

```
pixfirewall(config)# object-group network sjc_eng_ftp_servers
    pixfirewall(config-network)#network-object host sjc.eng.ftp.servcers
    pixfirewall(config-network)#network-object host 172.23.56.194
    pixfirewall(config-network)#network-object 192.1.1.0 255.255.255.224
    pixfirewall(config-network)#exit
```

The following example shows how to use the **object-group network** subcommand to create a new network object group and map it to a existing object-group:

```
pixfirewall(config)# object-group network sjc_ftp_servers
    pixfirewall(config-network)#network-object host sjc.ftp.servers
    pixfirewall(configpixfirewall(config-network)#network-object host 172.23.56.195
    pixfirewall(config-network)#network-object 193.1.1.0 255.255.255.224
    pixfirewall(config-network)#group-object sjc_eng_ftp_servers
    pixfirewall(config-network)#exit
```

The following example shows how to use the **object-group protocol** subcommand mode to create a new protocol object group:

```
pixfirewall(config)# object-group protocol proto_grp_1
    pixfirewall(config-protocol)#protocol-object udp
    pixfirewall(config-protocol)#protocol-object ipsec
    pixfirewall(config-protocol)#exit

pixfirewall(config)# object-group protocol proto_grp_2
    pixfirewall(config-protocol)#protocol-object tcp
    pixfirewall(config-protocol)#group-object proto_grp_1
    pixfirewall(config-protocol)#exit
```

The following example shows how to use the **object-group service** subcommand mode to create a new port (service) object group:

```
pixfirewall(config)# object-group service eng_service tcp
    pixfirewall(config-service)#group-object eng_www_service
    pixfirewall(config-service)#port-object eq ftp
    pixfirewall(config-service)#port-object range 2000 2005
    pixfirewall(config-service)#exit
```

The following example shows how to add and remove a text description to an object group:

```
pixfirewall(config)# object-group protocol protos1
    pixfirewall(config-protocol)# description This group of protocols is for our internal
    network
```

```
    pixfirewall(config-protocol)# show object-group id protos1
    object-group protocol protos1
      description: This group of protocols is for our internal network

    pixdocipsec1(config-protocol)# no description
    pixdocipsec1(config-protocol)# show object-group id protos1
    object-group protocol protos1
```

The following example shows how to use the **object groupinggroup-object** subcommand mode to create a new object group that consists of previously defined objects:

```
pixfirewall(config)# object-group network host_grp_1
    pixfirewall(config-network)# network-object host 192.168.1.1
    pixfirewall(config-network)# network-object host 192.168.1.2
    pixfirewall(config-network)# exit

pixfirewall(config)# object-group network host_grp_2
    pixfirewall(config-network)# network-object host 172.23.56.1
    pixfirewall(config-network)# network-object host 172.23.56.2
    pixfirewall(config-network)# exit

pixfirewall(config)# object-group network all_hosts
    pixfirewall(config-network)# group-object host_grp_1
    pixfirewall(config-network)# group-object host_grp_2
    pixfirewall(config-network)# exit

pixfirewall(config)# access-list grp_1 permit tcp object-group host_grp_1 any eq ftp
pixfirewall(config)# access-list grp_2 permit tcp object-group host_grp_2 any eq smtp
pixfirewall(config)# access-list all permit tcp object-group all_hosts any eq www
```

As shown in this example, without the **group-object** command the *all_hosts* group has to be defined to include all the IP addresses that have already defined in *host_grp_1* and *host_grp_2*, but with the **group-object** command, the duplicated definitions of the hosts are eliminated.

The following example illustrates how use object groups to simplify access list configuration:

```
object-group network remote
    network-object host kqk.suu.dri.ixx
    network-object host kqk.suu.pyl.gnl

object-group network locals
    network-object host 172.23.56.10
    network-object host 172.23.56.20
    network-object host 172.23.56.194
    network-object host 172.23.56.195

object-group service eng_svc ftp
    port-object eq www
    port-object eq smtp
    port-object range 25000 25100
```

This grouping then enables the access list to be configured in one line instead of 24 lines, which would be needed if no grouping is used. Instead, with the grouping, the access list configuration is as follows:

```
access-list acl permit tcp object-group remote object-group locals object-group eng_svc
```

Note The **show config** and **write** commands display the access list as configured with the object group names. However, the **show access-list** command displays the access list entries expanded out into individual statements without their object groupings.

outbound/apply

Create an access list for controlling Internet use.

[no] apply [(*if_name*)] *list_ID* **outgoing_src | outgoing_dest**

clear apply

[no] outbound *list_ID* **permit | deny** *ip_address* [*netmask* [*port*[**-***port*]] [*protocol*]

[no] outbound *list_ID* **except** *ip_address* [*netmask* [*port*[**-***port*]] [*protocol*]

clear outbound

show apply [(*if_name*)] [*list_ID* **outgoing_src | outgoing_dest**]

show outbound

Syntax Description

apply	Specifies whether the access control list applies to inside users' ability to start outbound connections with **apply** command's **outgoing_src** option, or whether the access list applies to inside users' ability to access servers on the outside network with the **apply** command's **outgoing_dest** option.
clear apply	Removes all the **apply** command statements from the configuration.
clear outbound	Removes all **outbound** command statements from the configuration.
deny	Deny the access list access to the specified IP address and port.
except	Create an exception to a previous **outbound** command. An **except** command statement applies to **permit** or **deny** command statements only with the same access list ID. When used with **apply outgoing_src**, the IP address of an **except** command statement applies to the destination address. When used with **apply outgoing_dest**, the IP address of an **except** command statement applies to the source address. See "Outbound List Rules" for more information.
if_name	The network interface originating the connection.
ip_address	The IP address for this access list entry. Do not specify a range of addresses. The 0.0.0.0 *ip_address* can be abbreviated as 0.
list_ID	A tag number for the access list. The access list number you use must be the same for the **apply** and **outbound** commands. This value must be a positive number from 1 to 1599. This number can be the same as what you use with the **nat** and **global** commands. This number is just an arbitrary number that groups **outbound** command statements to an **apply** command statement. *List_IDs* are processed sequentially in descending order. For more information, see "Outbound List Rules."
netmask	The network mask for comparing with the IP address; 255.255.255.0 causes the access list to apply to an entire Class C address. 0.0.0.0 indicates all access. The 0.0.0.0 *netmask* can be abbreviated as 0.
no outbound	Removes a single **outbound** command statement from the configuration.

no apply	Removes a single **apply** command statement from the configuration.
outbound	The **outbound** command, in conjunction with the **apply** command, uses access lists to control a filtering function on outgoing packets from the PIX Firewall. The filters can be based on the source IP address, the destination IP address, and the destination port/protocol as specified by the rules. The use of an **outbound** command requires use of the **apply** command. The **apply** command lets you specify whether the access control list applies to inside users' ability to start outbound connections with the **apply** command's **outgoing_src** option, or whether the access list applies to inside users' ability to access servers on the outside network with the **apply** command's **outgoing_dest** option. For more information, see "Outbound List Rules" and the **access-list** command. The **outbound** command has been superseded by the **access-list** command.
outgoing_dest	Deny or permit access to an external IP address using the service(s) specified in the **outbound** command.
outgoing_src	Deny or permit an internal IP address the ability to start outbound connections using the service(s) specified in the **outbound** command.
permit	Allow the access list to access the specified IP address and port.
port	A port or range of ports that the access list is permitted or denied access to. See the "Ports" section in Chapter 2, "Using PIX Firewall Commands" for a list of valid port literal names.
protocol	Limit outbound access to **udp**, **tcp**, or **icmp** protocols. If a protocol is not specified, the default is **tcp**.

Command Modes

Configuration mode.

Usage Guidelines

The **outbound** command creates an access list that lets you specify the following:

- Whether inside users can create outbound connections
- Whether inside users can access specific outside servers
- What services inside users can use for outbound connections and for accessing outside servers
- Whether outbound connections can execute Java applets on the inside network

Outbound lists are filters on outgoing packets from the PIX Firewall. The filter can be based on the source IP address, the destination IP address, and the destination port/protocol as specified by the rules. The use of an **outbound** command requires use of the **apply** command. The **apply** command enables you to specify whether the access control list applies to inside users' ability to start outbound connections with **apply** command's **outgoing_src** option, or whether the access list applies to inside users' ability to access servers on the outside network with the **apply** command's **outgoing_dest** option.

Note The **outbound** command has been superseded by the **access-list** command. We recommend that you migrate your **outbound** command statements to **access-list** command statements to maintain future compatibility.

The **java** option has been replaced by the **filter java** command.

After adding, removing, or changing **outbound** command statements, use the **clear xlate** command.

Use the **no outbound** command to remove a single **outbound** command statement from the configuration. Use the **clear outbound** command to remove all **outbound** command statements from the configuration. The **show outbound** command displays the **outbound** command statements in the configuration.

Use the **no apply** command to remove a single **apply** command statement from the configuration. Use the **clear apply** command statement to remove all the **apply** command statements from the configuration. The **show apply** command displays the **apply** command statements in the configuration.

Outbound List Rules

Rules, written as **outbound** *list_ID* command statements are global to the PIX Firewall; they are activated by **apply** *list_ID outgoing_src* | *outgoing_dest* command statements. When applied to *outgoing_src*, the source IP address, the destination port, and protocol are filtered. When applied to *outgoing_dest*, the destination IP address, port, and protocol are filtered.

The *outgoing_src* option and *outgoing_dest* outbound lists are filtered independently. If any one of the filters contain the **deny** option, the outbound packet is denied. When multiple rules are used to filter the same packet, the best matched rule takes effect. The best match is based on the IP address mask and the port range check. More strict IP address masks and smaller port ranges are considered a better match. If there is a tie, a **permit** option overrides a **deny** option.

Rules are grouped by a *list_ID*. Within each *list_ID*, **except** rules (that is, **outbound** *n* **except** …) can be set. The **except** option reverses the best matched rule of **deny** or **permit**. In addition, PIX Firewall filters the specified IP address and mask in the rule for the destination IP address of the outbound packet if the list is applied to the *outbound_src*. Alternatively, PIX Firewall filters the source IP address if the list is applied to the *outgoing_dest*. Furthermore, the **except** rules only apply to rules with the same *list_ID*. A single **except** rule within a *list_ID* without another **permit** or **deny** rule has no effect. If multiple **except** rules are set, the best match is checked for which **except** to apply.

The **outbound** command rules are now sorted by the best match checking. Use the **show outbound** command to see how the best match is judged by the PIX Firewall.

Usage Notes

1. If **outbound** commands are not specified, the default behavior is to permit all outbound traffic and services from inside hosts.
2. After adding, changing, or removing an **outbound** and **apply** command statement group, use the **clear xlate** command to make the IP addresses available in the translation table.
3. The **outbound** commands are processed linearly within a *list_ID*. In addition, *list_ID*s are processed sequentially in descending order. For example, the first command statement you specify in an **outbound** list is processed first, then the next **outbound** command statement in that list, and so on. Similarly, *list_ID* 10 is processed before *list_ID* 20, and so on.
4. When using **outbound** commands, it is often helpful to deny or permit access to the many before you deny or permit access to the specific. Start with an interface-wide specification such as the following command that denies all hosts from starting connections.

```
outbound 1 deny 0 0 0
apply (inside) 1 outgoing_src
```

Then add command statements that permit or deny hosts access to specific ports.

For example:

```
outbound 1 deny 0 0 0
outbound 1 permit 10.1.1.1 255.255.255.255 23 tcp
outbound 1 permit 10.1.1.1 255.255.255.255 80 tcp
apply (inside) 1 outgoing_src
```

You could state this same example as follows with the **except** option:

```
outbound 1 deny 0 0 0
outbound 1 except 209.165.201.11 255.255.255.255 23 tcp
outbound 1 except 209.165.201.11 255.255.255.255 80 tcp
apply (inside) 1 outgoing_src
```

In the preceding **outbound except** command statement, IP address 209.165.201.11 is the destination IP address, not the source address. This means that everyone is denied outbound access, except those users going to 209.165.201.11 via Telnet (port 23) or HTTP (port 80).

5. If you permit access to port 80 (**http**), this also permits Java applets to be downloaded. You must have a specific **deny** command statement to block Java applets.
6. The maximum number of **outbound** list entries in a configuration is 1599.
7. Outbound lists have no effect on **access-list** command statement groups.
8. The use of the **access-group** command statement overrides the **conduit** and **outbound** command statements for the specified interface name.

Examples

In the following example, the first **outbound** group sets inside hosts so that they can only see and Telnet to perimeter hosts, and do DNS lookups. The perimeter network address is 209.165.201.0 and the network mask is 255.255.255.224.

```
outbound 9 deny 0.0.0.0 0.0.0.0 0 0
outbound 9 except 209.165.201.0 255.255.255.224 23 tcp
outbound 9 except 0.0.0.0 0.0.0.0 53 udp
```

The next **outbound** group lets hosts 10.1.1.11 and 10.1.1.12 go anywhere:

```
outbound 11 deny 0.0.0.0 0.0.0.0 0 0
outbound 11 permit 10.1.1.11 255.255.255.255 0 0
outbound 11 permit 10.1.1.12 255.255.255.255 0 0
outbound 11 permit 0.0.0.0 0.0.0.0 21 tcp
outbound 11 permit 10.3.3.3 255.255.255.255 143 tcp
```

This last **outbound** group lets hosts on the perimeter only access TCP ports 389 and 30303 and UDP port 53 (DNS).

Note The PIX Firewall drops DNS packets sent to UDP port 53 that have a packet size larger than 512 bytes.

Finally, the **apply** command statements set the **outbound** groups so that the permit and deny rules affect access to all external addresses.

```
outbound 13 deny 0.0.0.0 0.0.0.0 0 0
outbound 13 permit 0.0.0.0 0.0.0.0 389 tcp
outbound 13 permit 0.0.0.0 0.0.0.0 30303 tcp
outbound 13 permit 0.0.0.0 0.0.0.0 53 udp

apply (inside) 9 outgoing_src
apply (inside) 11 outgoing_src
apply (perim) 13 outgoing_src
```

Controlling Outbound Connections

The following example prevents all inside hosts from starting outbound connections:

```
outbound 1 deny 0 0 0
apply (inside) 1 outgoing_src
```

The **0 0 0** at the end of the command means all IP addresses (**0** is the same as **0.0.0.0**), with a 0.0.0.0 subnet mask and for all services (port value is zero).

Conversely, the following example permits all inside hosts to start connections to the outside (this is the default if an access list is not created):

```
outbound 1 permit 0 0 0
apply (inside) 1 outgoing_src
```

Controlling Inside Hosts' Access to Outbound Services

The following example prevents inside host 192.168.1.49 from accessing the World Wide Web (port 80):

```
outbound 11 deny 192.168.1.49 255.255.255.255 80 tcp
apply (inside) 11 outgoing_src
```

Controlling Inside Hosts' Access to Outside Servers

If your employees are spending too much time examining GIF images on a particular website with two web servers, you can use the following example to restrict this access:

```
outbound 12 deny 192.168.146.201 255.255.255.255 80 tcp
outbound 12 deny 192.168.146.202 255.255.255.255 80 tcp
apply (inside) 12 outgoing_dest
```

Using except Command Statements

An **except** command statement only provides exception to items with the same *list_ID*, as shown in the following example:

```
outbound 9 deny 0.0.0.0 0.0.0.0 0 0
outbound 9 except 10.100.0.0 255.255.0.0 23 tcp
outbound 9 except 0.0.0.0 0.0.0.0 53 udp
outbound 11 deny 0.0.0.0 0.0.0.0 0 0
outbound 11 permit 10.1.1.11 255.255.255.255 0 0
outbound 11 permit 10.1.1.12 255.255.255.255 0 0
outbound 11 permit 0.0.0.0 0.0.0.0 21 tcp
outbound 11 permit 10.3.3.3 255.255.255.255 143 tcp
outbound 13 deny 0.0.0.0 0.0.0.0 0 0
outbound 13 permit 0.0.0.0 0.0.0.0 389 tcp
outbound 13 permit 0.0.0.0 0.0.0.0 30303 tcp
outbound 13 permit 0.0.0.0 0.0.0.0 53 udp
```

In the preceding examples, the following two command statements work against other command statements in list 9 but not in lists 11 and 13:

```
outbound 9 except 10.100.0.0 255.255.0.0 23 tcp
outbound 9 except 0.0.0.0 0.0.0.0 53 udp
```

In the following example, the set of **deny**, **permit**, and **except** option command statements denies everybody from connecting to external hosts except for DNS queries and Telnet connections to hosts on 10.100.0.0. The host with IP address 10.1.1.11 is permitted outbound access, and has access to everywhere *except* to 10.100.0.0 via Telnet and anywhere to use DNS.

```
outbound   1 deny 0.0.0.0 0.0.0.0 0 tcp
outbound   1 permit 10.1.1.11 255.255.255.255 0 tcp
outbound   1 except 10.100.0.0 255.255.0.0 23 tcp
outbound   1 except 0.0.0.0 0.0.0.0 53 udp
apply (inside) outgoing_src
```

pager

Enable or disable screen paging.

[**no**] **pager** [**lines** *number*]

clear pager

show pager

Syntax Description

number	The number of lines before the "---more---" prompt appears. The minimum is **1**. Use **0** to disable paging.

Command Modes

Privileged mode.

Usage Guidelines

The **pager lines** command let you specify the number of lines in a page before the "---more---" prompt appears. The **pager** command enables display paging, and the **no pager** command disables paging and lets output display completely without interruption. If you set the **pager lines** command to some value and want to revert back to the default, enter the **pager** command without options. The **clear pager** command resets the number of lines in a page to 24.

When paging is enabled, the following prompt appears:

```
<--- more --->
```

The "---more---" prompt uses syntax similar to the UNIX **more** command:

- To view another screenful, press the Space bar.
- To view the next line, press the **Enter** key.
- To return to the command line, press the **q** key.

Use the **pager 0** command to disable paging.

Examples

The following example shows use of the **pager** command:

```
pixfirewall# pager lines 2
pixfirewall# ping inside 10.0.0.42
        10.0.0.42 NO response received -- 1010ms
        10.0.0.42 NO response received -- 1000ms
<--- more --->
```

password

Set password for Telnet access to the PIX Firewall console.

{**password** | **passwd**} *password* [**encrypted**]

clear {**password** | **passwd**}

show {**password** | **passwd**}

Syntax Description

encrypted	Specifies that the password you entered is already encrypted. The *password* you specify with the **encrypted** option must be 16 characters in length.
password	A case-sensitive password of up to 16 alphanumeric and special characters. Any character can be used in the password except a question mark and a space.

Command Modes

Privileged and configuration modes.

Usage Guidelines

The **password** command sets a password for Telnet access to the PIX Firewall console. The keyword **passwd** is also accepted as a shortened form of **password**. Additionally, the firewall configuration displays the password using the short form, **passwd.**

An empty password is changed into an encrypted string. However, any use of a **write** command displays or writes the passwords in encrypted form. Once passwords are encrypted, they are not reversible back to plain text. The **clear password** command resets the password to "cisco."

Note Write down the new password and store it in a manner consistent with your site's security policy. Once you change this password, you cannot view it again.

The **show password** command displays the Telnet password.

Examples

The following example shows use of the **password** command:

```
pixfirwall(config)# password watag00s1am
pixfirwall(config)# show passwd
passwd jMorNbK0514fadBh encrypted
```

Related Commands		
	enable	Configures enable passwords.
	telnet	Adds Telnet access to the firewall console and sets the idle timeout.

pdm

These commands support communication between the PIX Firewall and a browser running the Cisco PIX Device Manager (PDM).

show pdm sessions

pdm disconnect *session_id*

pdm history enable

pdm history [view {all | 12h | 5d | 60m | 10m}] [snapshot] [feature {all | blocks | cpu | failover | ids | interface *if_name* **| memory | perfmon | xlates}] [pdmclient]**

pdm group *real_group_name associated_intf_name*

pdm group *ref_group_name ref_intf_name* **reference** *real_group_name*

pdm location ***ip_address netmask if_name***

pdm logging [*level* **[***messages***]]**

show pdm history

show pdm logging

show pdm sessions

clear pdm

Syntax Description	
12h | 5d | 60m | 10m | all	Specifies the PDM history view to display: 12 hours (**12h**), 5 days (**5d**), 60 minutes (**60m**),10 minutes (**10m**), or **all** history contents in the PDM history buffer.
associated_intf_name	The name of the interface to which the specified object group is associated. This name must have been defined by the **nameif** command.
blocks	History for system buffers. Similar to output from the **show blocks** command.
clear pdm	Removes all locations, disables logging, and clears the PDM buffer. Internal PDM command.
cpu	History for CPU usage. Similar to output from the **show cpu usage** command.
failover	History for failover. Similar to output from the **show failover** command.
feature	This specifies to display history for a single feature (selected with one of the following). Otherwise, all of them are displayed.

history enable	Internal PDM command. Take a data sample and store the sample data to the PDM history buffer. The **no** version of this command disables PDM data sampling.
ids	History for IDS (Intrusion Detection System).
if_name	Specifies the interface name on which PDM resides.
ip_address	Specifies the host or network on which PDM resides.
level	Specifies the priority level of syslog messages displayed in the PDM **syslog** option.
location	Assists PDM with network topology discovery by associating an external network object with an interface. Note: The **pdm location** command does not control which host can launch PDM. See [**no**] **http** *ip_address* [*netmask*] [*if_name*] for this function.
logging	Internal PDM command. Specifies the type and number of syslog messages displayed through the PDM **syslog** option.
memory	History for memory. Similar to output from the **show memory** command.
messages	Specifies the number of messages stored in the PDM buffer. Once the buffer is full, old messages will be discarded.
netmask	Specifies the network mask for the **pdm location** *ip_address*.
pdm	Specifies the Cisco PIX Device Manager.
pdm disconnect	Disconnects the specified PDM session from the PIX Firewall.
pdmclient	Displays the PDM history in PDM-display format.
perfmon	History for performance. Similar to output from the **show perfmon** command.
session_id	PDM session ID number available from the **show pdm sessions** command.
snapshot	Displays only the last PDM history data point.
real_group_name	The name of a PDM object group that contains real IP addresses.
ref_group_name	The name of an object group which contains network address translated (NATed) IP addresses of the object group specified by *real_group_name*.
ref_intf_name	The name of the interface from which the destination IP address of inbound traffic is network address translated (NATed). This name must have been defined by the **nameif** command.
xlates	History for translation slot information. Similar to output from the **show xlate** command.

Defaults

Default PDM syslog *level* is **0**. Default logging *messages* is **100** and the maximum is **512**.

Command Modes

Configuration mode.

Usage Guidelines

The **pdm disconnect** command and the **show pdm sessions** commands are accessible through the PIX Firewall command-line interface (CLI). The **show pdm sessions** command lists all the active PDM sessions connected to the PIX Firewall by a unique se*ssion_id*, beginning with session number **0**. The **pdm disconnect** command lets you disconnect a specific PDM session using its *session_id*.

The **show pdm history** command displays the contents of the PDM history buffer.

The **show pdm logging** command displays the contents of the PDM logging buffer (located within PDM). PDM syslog messages are stored separately from the PIX Firewall syslog messages. The **clear pdm logging** command clears the PDM log without disabling PDM logging.

The **clear pdm, pdm group, pdm history**, **pdm location, and pdm logging** commands may appear in your configuration, but they are designed to work as internal PDM-to-PIX Firewall commands accessible only to PDM.

The **pdm location** command associates an interface to an *ip_address /netmask* pair. Specifying a new pair replaces the old definition. The **clear pdm location** command removes all of the PDM locations.

Note Note: The **pdm location** command does not control which host can launch PDM. See [**no**] **http** *ip_address* [*netmask*] [*if_name*] for this function.

PDM location is not actually a PIX command, but rather a PDM bookkeeping command. When PDM opens it discovers the network topology surrounding the PIX from which it was launched. PDM then stores its discovered topology database in the PIX config file using **pdm location** commands to record ip address to interface associations. For example:

```
pdm location 10.1.1.1 255.255.255.255 inside
pdm location 10.1.1.2 255.255.255.255 inside
pdm location 10.1.3.0 255.255.255.0 inside
pdm location 10.1.2.0 255.255.255.0 outside
pdm location InsideRouter 255.255.255.255 inside
```

PDM rules are built on top of the network topology it can discover or has explicitly defined. Ideally, the topology is clearly defined first via the Host/Network and Network Object functions before policy Rules are applied.

You may use the CLI command **clear pdm location** to remove **pdm location** commands from your configuration, and it will not affect the operation of the PIX. However the next time PDM is run, it will again have to rediscover the network topology and update the configuration file with **pdm location** commands.

If you have an existing configuration before migrating to PDM, or use both the CLI and PDM to configure your PIX Firewall, PDM will derive much of the topology information from the current config file. For example:

```
static (inside,outside) 2.2.2.2 1.1.1.2 netmask 255.255.255.255 0 0
```

This command implies that **host 1.1.1.2** resides on the **inside** network.

Why is **pdm location** needed if PDM can derive or discover the topology information at runtime?

- The **static** command can be removed. If the location of **1.1.1.2** is not defined elsewhere in the config, the interface association will not be available to PDM. This can happen if you implicitly changed topology while editing an Access Rule or Translation Rule.
- PDM may not be able to resolve all the IP addresses shown in a configuration. For example, a PIX with three interfaces uses the CLI command **acl permit ip any 1.1.1.1** applied to **inside** interface. Where is **1.1.1.1, dmz** or **outside**? If you manually resolve **1.1.1.1** to the **outside** interface, for example, PDM will need to “remember” the interface to IP address association to allow Rules to be accurately displayed and edited.

The following example shows how to report the last data point in PDM-display format:

```
pix(config)# pdm history enable
pix(config)# show pdm history view 10m snapshot pdmclient
INTERFACE|outside|up|IBC|0|OBC|1088|IPC|0|OPC|0|IBR|17|OBR|
0|IPR|0|OPR|0|IERR|1|NB|0|RB|0|RNT|0|GNT|0|CRC|0|FRM|0|OR|
0|UR|0|OERR|0|COLL|0|LCOLL|0|RST|0|DEF|0|LCR|
0:PIXoutsideINTERFACE:METRIC_HISTORY|SNAP|IBR|VIEW|10|1952|
METRIC_HISTORY|SNAP|OBR|VIEW|10|64|METRIC_HISTORY|SNAP|IPR|
VIEW|10|17|METRIC_HISTORY|SNAP|OPR|VIEW|10|1|METRIC_HISTORY|
SNAP|IERR|VIEW|10|0|METRIC_HISTORY|SNAP|OERR|VIEW|10|0|
:PIXinsideINTERFACE:METRIC_HISTORY|SNAP|IBR|VIEW|10|0|
METRIC_HISTORY|SNAP|OBR|VIEW|10|64|METRIC_HISTORY|SNAP|IPR|
VIEW|10|0|METRIC_HISTORY|SNAP|OPR|VIEW|10|1|METRIC_HISTORY|
SNAP|IERR|VIEW|10|0|METRIC_HISTORY|SNAP|OERR|VIEW|10|0|
:PixSYS:METRIC_HISTORY|SNAP|MEM|VIEW|10|52662272|
METRIC_HISTORY|SNAP|BLK4|VIEW|10|1600|METRIC_HISTORY|SNAP|
BLK80|VIEW|10|400|METRIC_HISTORY|SNAP|BLK256|VIEW|10|998|
METRIC_HISTORY|SNAP|BLK1550|VIEW|10|676|METRIC_HISTORY|SNAP|
XLATES|VIEW|10|0|METRIC_HISTORY|SNAP|CONNS|VIEW|10|0|
METRIC_HISTORY|SNAP|TCPCONNS|VIEW|10|0|METRIC_HISTORY|SNAP|
UDPCONNS|VIEW|10|0|METRIC_HISTORY|SNAP|URLS|VIEW|10|0|
METRIC_HISTORY|SNAP|WEBSNS|VIEW|10|0|METRIC_HISTORY|SNAP|
TCPFIXUPS|VIEW|10|0|METRIC_HISTORY|SNAP|TCPINTERCEPTS|VIEW|
10|0|METRIC_HISTORY|SNAP|HTTPFIXUPS|VIEW|10|0|METRIC_HISTORY|
SNAP|FTPFIXUPS|VIEW|10|0|METRIC_HISTORY|SNAP|AAAAUTHENUPS|VIEW|
10|0|METRIC_HISTORY|SNAP|AAAAUTHORUPS|VIEW|10|0|METRIC_HISTORY|
SNAP|AAAACCOUNTS|VIEW|10|0|
```

The following example shows how to report the data, formatted for the PIX Firewall CLI:

```
pix(config)# pdm history enable
pix(config)# show pdm history view 10m snapshot
Available 4 byte Blocks:  [  10s] : 1600
Used 4 byte Blocks:  [  10s] : 0
Available 80 byte Blocks:  [  10s] : 400
Used 80 byte Blocks:  [  10s] : 0
Available 256 byte Blocks:  [  10s] : 500
Used 256 byte Blocks:  [  10s] : 0
Available 1550 byte Blocks:  [  10s] : 931
Used 1550 byte Blocks:  [  10s] : 385
Available 1552 byte Blocks:  [  10s] : 0
Used 1552 byte Blocks:  [  10s] : 0
Available 2560 byte Blocks:  [  10s] : 0
Used 2560 byte Blocks:  [  10s] : 0
Available 4096 byte Blocks:  [  10s] : 0
Used 4096 byte Blocks:  [  10s] : 0
Available 8192 byte Blocks:  [  10s] : 0
Used 8192 byte Blocks:  [  10s] : 0
Available 16384 byte Blocks:  [  10s] : 0
Used 16384 byte Blocks:  [  10s] : 0
Available 65536 byte Blocks:  [  10s] : 0
Used 65536 byte Blocks:  [  10s] : 0
CPU Utilization:  [  10s] : 0
IP Options Bad:  [  10s] : 0
Record Packet Route:  [  10s] : 0
IP Options Timestamp:  [  10s] : 0
Provide s,c,h,tcc:  [  10s] : 0
Loose Source Route:  [  10s] : 0
SATNET ID:  [  10s] : 0
Strict Source Route:  [  10s] : 0
IP Fragment Attack:  [  10s] : 0
Impossible IP Attack:  [  10s] : 0
IP Teardrop:  [  10s] : 0
```

```
ICMP Echo Reply:  [  10s] : 0
ICMP Unreachable:  [  10s] : 0
ICMP Source Quench:  [  10s] : 0
ICMP Redirect:  [  10s] : 0
ICMP Echo Request:  [  10s] : 0
ICMP Time Exceeded:  [  10s] : 0
ICMP Parameter Problem:  [  10s] : 0
ICMP Time Request:  [  10s] : 0
ICMP Time Reply:  [  10s] : 0
ICMP Info Request:  [  10s] : 0
ICMP Info Reply:  [  10s] : 0
ICMP Mask Request:  [  10s] : 0
ICMP Mask Reply:  [  10s] : 0
Fragmented ICMP:  [  10s] : 0
Large ICMP:  [  10s] : 0
Ping of Death:  [  10s] : 0
No Flags:  [  10s] : 0
SYN & FIN Only:  [  10s] : 0
FIN Only:  [  10s] : 0
FTP Improper Address:  [  10s] : 0
FTP Improper Port:  [  10s] : 0
Bomb:  [  10s] : 0
Snork:  [  10s] : 0
Chargen:  [  10s] : 0
DNS Host Info:  [  10s] : 0
DNS Zone Transfer:  [  10s] : 0
DNS Zone Transfer High Port:  [  10s] : 0
DNS All Records:  [  10s] : 0
Port Registration:  [  10s] : 0
Port Unregistration:  [  10s] : 0
RPC Dump:  [  10s] : 0
Proxied RPC:  [  10s] : 0
ypserv Portmap Request:  [  10s] : 0
ypbind Portmap Request:  [  10s] : 0
yppasswd Portmap Request:  [  10s] : 0
ypupdated Portmap Request:  [  10s] : 0
ypxfrd Portmap Request:  [  10s] : 0
mountd Portmap Request:  [  10s] : 0
rexd Portmap Request:  [  10s] : 0
rexd Attempt:  [  10s] : 0
statd Buffer Overflow:  [  10s] : 0
Input KByte Count:  [  10s] : 41804
Output KByte Count:  [  10s] : 526456
Input KPacket Count:  [  10s] : 364
Output KPacket Count:  [  10s] : 450
Input Bit Rate:  [  10s] : 0
Output Bit Rate:  [  10s] : 0
Input Packet Rate:  [  10s] : 0
Output Packet Rate:  [  10s] : 0
Input Error Packet Count:  [  10s] : 0
No Buffer:  [  10s] : 0
Received Broadcasts:  [  10s] : 90076
Runts:  [  10s] : 0
Giants:  [  10s] : 0
CRC:  [  10s] : 0
Frames:  [  10s] : 0
Overruns:  [  10s] : 0
Underruns:  [  10s] : 0
Output Error Packet Count:  [  10s] : 0
Collisions:  [  10s] : 8895
LCOLL:  [  10s] : 0
Reset:  [  10s] : 0
Deferred:  [  10s] : 3138
Lost Carrier:  [  10s] : 0
```

```
Hardware Input Queue:  [  10s] : 128
Software Input Queue:  [  10s] : 0
Hardware Output Queue:  [  10s] : 0
Software Output Queue:  [  10s] : 0
Input KByte Count:  [  10s] : 61835
Output KByte Count:  [  10s] : 26722
Input KPacket Count:  [  10s] : 442
Output KPacket Count:  [  10s] : 418
Input Bit Rate:  [  10s] : 0
Output Bit Rate:  [  10s] : 0
Input Packet Rate:  [  10s] : 0
Output Packet Rate:  [  10s] : 0
Input Error Packet Count:  [  10s] : 0
No Buffer:  [  10s] : 0
Received Broadcasts:  [  10s] : 308607
Runts:  [  10s] : 0
Giants:  [  10s] : 0
CRC:  [  10s] : 0
Frames:  [  10s] : 0
Overruns:  [  10s] : 0
Underruns:  [  10s] : 0
Output Error Packet Count:  [  10s] : 0
Collisions:  [  10s] : 0
LCOLL:  [  10s] : 0
Reset:  [  10s] : 0
Deferred:  [  10s] : 2
Lost Carrier:  [  10s] : 707
Hardware Input Queue:  [  10s] : 128
Software Input Queue:  [  10s] : 0
Hardware Output Queue:  [  10s] : 0
Software Output Queue:  [  10s] : 0
Available Memory:  [  10s] : 45293568
Used Memory:  [  10s] : 21815296
Xlate Count:  [  10s] : 0
Connection Count:  [  10s] : 0
TCP Connection Count:  [  10s] : 0
UDP Connection Count:  [  10s] : 0
URL Filtering Count:  [  10s] : 0
URL Server Filtering Count:  [  10s] : 0
TCP Fixup Count:  [  10s] : 0
TCP Intercept Count:  [  10s] : 0
HTTP Fixup Count:  [  10s] : 0
FTP Fixup Count:  [  10s] : 0
AAA Authentication Count:  [  10s] : 0
AAA Authorzation Count:  [  10s] : 0
AAA Accounting Count:  [  10s] : 0
Current Xlates:  [  10s] : 0
Max Xlates:  [  10s] : 0
ISAKMP SAs:  [  10s] : 0
IPSec SAs:  [  10s] : 0
L2TP Sessions:  [  10s] : 0
L2TP Tunnels:  [  10s] : 0
PPTP Sessions:  [  10s] : 0
PPTP Tunnels:  [  10s] : 0
```

Related Commands

setup	Preconfigures the firewall through interactive prompts.

perfmon

View performance information.

perfmon verbose

perfmon interval seconds

perfmon quiet

perfmon settings

show perfmon

Syntax Description

interval *seconds*	Specify the number of seconds the performance display is refreshed on the console. The default is 120 seconds.
quiet	Disable performance monitor displays.
settings	Displays the interval and whether it is quiet or verbose.
verbose	Enable displaying performance monitor information at the PIX Firewall console.

Command Modes

Privileged mode.

Usage Guidelines

The **perfmon** command lets you monitor the PIX Firewall unit's performance. Use the **show perfmon** command to view the information immediately. Use the **perfmon verbose** command to display the information every two minutes continuously. Use the **perfmon interval** *seconds* command with the **perfmon verbose** command to display the information continuously every number of seconds you specify.

Use the **perfmon quiet** command to disable the display.

The **show perfmon** command displays PIX Firewall performance information. (However, this command output does not display in a Telnet console session.)

An example of the performance information follows:

PERFMON STATS:	Current	Average
Xlates	33/s	20/s
Connections	110/s	10/s
TCP Conns	50/s	42/s
WebSns Req	4/s	2/s
TCP Fixup	20/s	15/s
HTTP Fixup	5/s	5/s
FTP Fixup	7/s	4/s
AAA Authen	10/s	5/s
AAA Author	9/s	5/s
AAA Account	3/s	3/s

This information lists the number of translations, connections, Websense requests, address translations (called "fixups"), and AAA transactions that occur each second.

Examples

The following commands display the performance monitor statistics every 30 seconds on the PIX Firewall console:

```
perfmon interval 30
perfmon verbose
```

ping

Determine if other IP addresses are visible from the PIX Firewall.

ping [*if_name*] *ip_address*

Syntax Description

if_name	The internal or external network interface name. The address of the specified interface is used as the source address of the ping.
ip_address	The IP address of a host on the inside or outside networks.

Command Modes

Privileged mode.

Usage Guidelines

The **ping** command determines if the PIX Firewall has connectivity or if a host is available on the network. The command output shows if the response was received; that is, that a host is participating on the network. If a host is not responding, **ping** displays "NO response received." Use the **show interface** command to ensure that the PIX Firewall is connected to the network and is passing traffic.

If you want internal hosts to be able to ping external hosts, you must create an ICMP **access-list** command statement for echo reply; for example, to give ping access to all hosts, use the **access-list** *acl_grp* **permit icmp any any** command and bind the **access-list** command statement to the interface you want to test using an **access-group** command statement.

If you are pinging through PIX Firewall between hosts or routers, but the pings are not successful, use the **debug icmp trace** command to monitor the success of the ping. If pings are both inbound and outbound, they are successful.

The PIX Firewall **ping** command no longer requires an interface name. If an interface name is not specified, PIX Firewall checks the routing table to find the address you specify. You can specify an interface name to indicate through which interface the ICMP echo requests are sent.

An example of the usage follows:

```
ping 10.0.0.1
         10.0.0.1 response received -- 10ms
         10.0.0.1 response received -- 10ms
         10.0.0.1 response received -- 0ms
```

Or you can still enter the command specifying the interface:

```
ping outside 10.0.0.1
         10.0.0.1 response received -- 10ms
         10.0.0.1 response received -- 10ms
```

```
10.0.0.1 response received -- 0ms
```

Examples In the following example, the **ping** command makes three attempts to reach an IP address:

```
ping 192.168.42.54
        192.168.42.54 response received -- 0Ms
        192.168.42.54 response received -- 0Ms
        192.168.42.54 response received -- 0Ms
```

prefix-list

Configures a prefix list for Area Border Router (ABR) type 3 link-state advertisement (LSA) filtering (to be used in OSPF routing areas).

[**no**] **prefix-list** *list_name* [**seq** *seq_number*] {**permit** | **deny** *prefix* **/** *len*} [**ge** *min_value*] [**le** *max_value*]

[**no**] **prefix-list sequence-number**

prefix-list *list_name* **description** *text*

Syntax Description

/	A required separator between the *prefix* and *len* values.
deny	Denies access for a matching condition.
ge	Applies the *min_value* to the range specified.
le	Applies the *max_value* to the range specified.
len	The network length (in bits) of the network mask, from 0 to 32.
list_name	The name of the prefix list. The *list_name* and *seq_number* together must be less than 64 characters combined.
max_value	Specifies the greater value of a range (the "to" portion of the range description). Ranges values can be from 0 to 32.
min_value	Specifies the lesser value of a range (the "from" portion of the range description). Ranges values can be from 0 to 32.
permit	Permits access for a matching condition.
prefix	The network number.
seq *seq_number*	Specifies the sequence number for the prefix list entry, from 1 to 4294967295. However, the *list_name* and *seq_number* together must be less than 64 characters combined.
sequence-number	Enables the generation of sequence numbers for entries in an OSPF prefix list.
text	The text of the description, with a maximum of 80 characters.

Defaults None.

Command Modes Configuration mode.

Usage Guidelines

The **prefix-list** commands are Area Border Router (ABR) type 3 link-state advertisement (LSA) filtering commands. ABR type 3 LSA filtering extends the capability of an ABR that is running OSPF to filter type 3 LSAs between different OSPF areas. This filtering is based on a prefix list defined by you, using the **prefix-list** commands. Once configured, only the specified prefixes are sent from one area to another area, and all other prefixes are restricted to their OSPF area. This type of area filtering can be applied to traffic going into or coming out of an OSPF area, or to both the incoming and outgoing traffic for that area.

To create an entry in a prefix list, use the **prefix-list** *list_name* command. To delete the entry, use the **no prefix-list** *list_name* command.

Use the **prefix-list** *list_name* **description** *text* command to add a text description to the prefix list name. To remove the text description, use the **no prefix-list** *list_name* **description** *text* command.

The **prefix-list** *list_name* **seq** *seq_number* command designates sequence numbers for entries in a prefix list.

Use the **prefix-list sequence-number** command to enable the generation of sequence numbers for entries in a OSPF prefix list.

Examples

The following example shows how to configure a prefix list:

```
pixfirewall(config)# prefix-list t-prelist permit 5/0001
pixfirewall(config)# show prefix-list
prefix-list t-prelist seq 5 permit 0.0.0.0/1
```

Related Commands

area filter-list	A subcommand to the **router ospf** command that uses the prefix list that you configure with the **prefix-list** command.
route-map	Creates a route map for redistributing routes from one routing protocol to another. Used in configuring OSPF routing on the firewall.
router ospf	Configures global parameters for the OSPF routing processes on the firewall, and enables or disables OSPF routing through the firewall.
routing interface	Configures interface-specific OSPF routing parameters.

privilege

Configures or displays command privilege levels.

[no] privilege [show | clear | configure] level *level* **[mode** *enable | configure***] command** *command*

show curpriv

show privilege [all | command *command* **| level** *level***]**

Syntax Description

clear	Sets the privilege level for the **clear** command corresponding to the command specified.
command	The command to allow. (Use the **no** command form to disallow.)
command	The command on which to set the privilege level.

configure	Sets the privilege level for the **configure** command corresponding to the command specified.
configure	For commands with both enable and configure modes, this indicates that the level is for the configure mode of the command.
curpriv	Displays the current privilege level.
detail	Displays privilege debugging information.
enable	For commands with both enable and configure modes, this indicates that the level is for the enable mode of the command.
level	The privilege level, from 0 to 15. (Lower numbers are lower privilege levels.)
level	Specifies the privilege level.
show	Sets the privilege level for the **show** command corresponding to the command specified.

Command Modes

Configuration mode.

Usage Guidelines

The privilege command sets user-defined privilege levels for PIX Firewall commands. This is especially useful for setting different privilege levels for related configuration, show, and clear commands. However, be sure to verify privilege level changes in your commands with your security policies before implementing the new privilege levels.

When commands have privilege levels set, and users have privilege levels set, then the two are compared to determine if a given user can execute a given command. If the user's privilege level is lower than the privilege level of the command, the user is prevented from executing the command. This is modeled after Cisco IOS software.

To change between privilege levels, use the **login** command to access another privilege level and the appropriate **logout**, **exit**, or **quit** command to exit that level.

Note Your **aaa authentication** and **aaa authorization** commands need to include any new privilege levels you define before you can use them in your AAA server configuration.

The **show curpriv** command displays the current privileges for a user.

The **show privilege** [**all** | **command** *command* | **level** *level*] command displays the privileges for a command or set of commands.

Examples

You can set the privilege level “5” for an individual user as follows:

```
username intern1 password pass1 privilege 5
```

You can also define a set of **show** commands with the privilege level “5” as follows:

```
level:

privilege show level 5 command alias
privilege show level 5 command apply
privilege show level 5 command arp
privilege show level 5 command auth-prompt
privilege show level 5 command blocks
```

The following examples show output from the **show curpriv** command when a user named **enable_15** is at different privilege levels. **Username** indicates the name the user entered when he or she logged in, **P_PRIV** indicates that the user has entered the **enable** command, and **P_CONF** indicates the user has entered the **config terminal** command.

```
pixfirewall(config)# show curpriv
Username : enable_15
Current privilege level : 15
Current Mode/s : P_PRIV P_CONF
pixfirewall(config)# exit

pixfirewall# show curpriv
Username : enable_15
Current privilege level : 15
Current Mode/s : P_PRIV
pixfirewall# exit

pixfirewall> show curpriv
Username : enable_1
Current privilege level : 1
Current Mode/s : P_UNPR
pixfirewall>
```

The following is an example of applying a privilege level of 11 to a complete AAA authorization configuration:

```
privilege configure level 11 command aaa
privilege configure level 11 command aaa-server
privilege configure level 11 command access-group
privilege configure level 11 command access-list
privilege configure level 11 command activation-key
privilege configure level 11 command age
privilege configure level 11 command alias
privilege configure level 11 command apply
```

Related Commands

aaa authentication	Enable, disable, or view LOCAL, TACACS+, or RADIUS user authentication, on a server designated by the **aaa-server** command, or PDM user authentication
login	Logs into a new privilege level.
object-group	Create an object group for use in other commands, such as **access-list** statements.
username	Configures local user authentication database.

quit

Exit configuration or privileged mode.

quit

Syntax Description

quit	Exits the current privilege level or mode.

Command Modes All modes.

Usage Guidelines Use the **quit** command to exit configuration or privileged mode.

Examples The following example shows use of the **quit** command:

```
pixfirewall(config)# quit
pixfirewall# quit
pixfirewall>
```

reload

Reboot and reload the configuration.

reload [**noconfirm**]

Syntax Description

noconfirm	Permits the PIX Firewall to reload without user confirmation.
reload	Reboot and reload configuration.

Command Modes Privileged mode.

Usage Guidelines The **reload** command reboots the PIX Firewall and reloads the configuration from a bootable floppy disk or, if a diskette is not present, from Flash memory.

The PIX Firewall does not accept abbreviations to the keyword **noconfirm**.

You are prompted for confirmation before starting with "Proceed with reload?".
Any response other than **n** causes the reboot to occur.

Note Configuration changes not written to Flash memory are lost after reload. Before rebooting, store the current configuration in Flash memory with the **write memory** command.

Examples The following example shows use of the **reload** command:

```
reload
Proceed with reload?  [confirm] y

Rebooting...

PIX Bios V2.7
...
```

rip

Change Routing Information Protocol (RIP) settings.

[no] rip *if_name* **default | passive** [**version** [**1 | 2**]] [**authentication** [**text | md5** *key* (*key_id*)]]

debug rip [*if_name*]

clear rip

show rip [*if_name*]

Syntax Description

authentication	Enable RIP Version 2 authentication.
default	Broadcast a default route on the interface.
if_name	The internal or external network interface name.
key	Key to encrypt RIP updates. This value must be the same on the routers and any other device *that provides RIP Version 2 updates*. The *key* is a text string of up to 16 characters in length.
key_id	Key identification value. The *key_id* can be a number from 1 to 255. Use the same *key_id* that is in use on the routers and any other device that provides RIP Version 2 updates.
md5	Send RIP updates using MD5 encryption.
passive	Enable passive RIP on the interface. The PIX Firewall listens for RIP routing broadcasts and uses that information to populate its routing tables.
text	Send RIP updates as clear text (not recommended).
version	RIP version. Use **version 2** for RIP update encryption. Use **version 1** to provide backward compatibility with the older version.

Command Modes

Configuration mode.

Usage Guidelines

The **rip** command enables IP routing table updates from received Routing Information Protocol (RIP) broadcasts. Use the **no rip** command to disable the PIX Firewall IP routing table updates. The default is to enable IP routing table updates. If you specify RIP Version 2, you can encrypt RIP updates using MD5 encryption.

The **clear rip** command removes all the **rip** commands from the configuration.

Ensure that the key and key_id values are the same as in use on any other device in your network that makes RIP Version 2 updates.

The PIX Firewall cannot pass RIP updates between interfaces.

When RIP Version 2 is configured in passive mode with PIX Firewall software Version 5.3 and higher, the PIX Firewall accepts RIP Version 2 multicast updates with an IP destination of 224.0.0.9. For RIP Version 2 default mode, the PIX Firewall will transmit default route updates using an IP destination of 224.0.0.9. Configuring RIP Version 2 registers the multicast address 224.0.0.9 on the respective interface to be able to accept multicast RIP Version 2 updates.

Only Intel 10/100 and Gigabit interfaces support multicasting.

When the RIP Version 2 commands for an interface are removed, the multicast address is unregistered from the interface card.

Examples

The following is sample output from the Version 1 **show rip** and **rip inside default** commands:

```
show rip
rip outside passive
no rip outside default
rip inside passive
no rip inside default

rip inside default
show rip
rip outside passive
no rip outside default
rip inside passive
rip inside default
```

The next example combines Version 1 and Version 2 commands and shows listing the information with the **show rip** command after entering the RIP commands that do the following:

- Enable Version 2 passive RIP using MD5 authentication on the outside interface to encrypt the key used by the PIX Firewall and other RIP peers, such as routers.
- Enable Version 1 passive RIP listening on the inside interface of the PIX Firewall.
- Enable Version 2 passive RIP listening on the **dmz** interface of the PIX Firewall.

```
rip outside passive version 2 authentication md5 thisisakey 2
rip outside default version 2 authentication md5 thisisakey 2
rip inside passive
rip dmz passive version 2

show rip
rip outside passive version 2 authentication md5 thisisakey 2
rip outside default version 2 authentication md5 thisisakey 2
rip inside passive version 1
rip dmz passive version 2
```

The next example shows how use of the **clear rip** command clears all the previous **rip** commands from the current configuration:

```
clear rip
show rip
```

The following example shows use of the Version 2 feature that passes the encryption key in text form:

```
rip out default version 2 authentication text thisisakey 3
show rip
rip outside default version 2 authentication text thisisakey 3
```

route

Enter a static or default route for the specified interface.

[**no**] **route** *if_name ip_address netmask gateway_ip* [*metric*]

clear route [*if_name ip_address* [*netmask gateway_ip*]]

show route

Syntax Description

gateway_ip	Specify the IP address of the gateway router (the next hop address for this route).
if_name	The internal or external network interface name.
ip_address	The internal or external network IP address. Use **0.0.0.0** to specify a default route. The **0.0.0.0** IP address can be abbreviated as **0**.
metric	Specify the number of hops to *gateway_ip*. If you are not sure, enter **1**. Your network administrator can supply this information or you can use a **traceroute** command to obtain the number of hops. The default is **1** if a metric is not specified.
netmask	Specify a network mask to apply to *ip_address*. Use **0.0.0.0** to specify a default route. The **0.0.0.0** *netmask* can be abbreviated as **0**.

Command Modes

Configuration mode.

Usage Guidelines

Use the **route** command to enter a default or static route for an interface. To enter a default route, set *ip_address* and *netmask* to **0.0.0.0,** or the shortened form of **0**. All routes entered using the **route** command are stored in the configuration when it is saved. The **clear route** command removes **route** command statements from the configuration that do not contain the CONNECT keyword.

Create static routes to access networks connected outside a router on any interface. The effect of a static route is like stating "to send a packet to the specified network, give it to this router." For example, PIX Firewall sends all packets destined to the 192.168.42.0 network through the 192.168.1.5 router with this static **route** command statement.

```
route dmz 192.168.42.0 255.255.255.0 192.168.1.5 1
```

The routing table automatically specifies the IP address of a PIX Firewall interface in the **route** command. Once you enter the IP address for each interface, PIX Firewall creates a **route** statement entry that is not deleted when you use the **clear route** command.

Note As of PIX Firewall Version 6.3(2), the **show route** command displays the route information only for active routes. Routes that are configured on interfaces and administratively or physically shut down do not display with the **show route** command.

If the **route** command statement uses the IP address from one interface of the PIX Firewall unit as the gateway IP address, PIX Firewall will ARP for the destination IP address in the packet instead of ARPing for the gateway IP address.

The following steps show how PIX Firewall handles routing:

Step 1 PIX Firewall receives a packet from the inside interface destined to IP address X.

Step 2 Because a default route is set to itself, PIX Firewall sends out an ARP for address X.

Step 3 Any Cisco router on the outside interface LAN which has a route to address X (Cisco IOS software has proxy ARP enabled by default) replies back to the PIX Firewall with its own MAC address as the next hop.

Step 4 PIX Firewall sends the packet to router (just like a default gateway).

Step 5 PIX Firewall adds the entry to its ARP cache for IP address X with the MAC address being that of the router.

- The CONNECT route entry is supported. (This identifier appears when you use the **show route** command.) The CONNECT identifier is assigned to an interface's local network and the interface IP address, which is in the IP local subnet. PIX Firewall will ARP for the destination address. The CONNECT identifier cannot be removed, but changes when you change the IP address on the interface.
- If you enter duplicate routes with different metrics for the same gateway, PIX Firewall changes the metric for that route and updates the metric for the route.

  For example, if the following command statement is in the configuration:

  ```
  route inside 10.0.0.0 255.0.0.0 10.0.0.2 2 OTHER
  ```

  If you enter the following statement:

  ```
  route inside 10.0.0.0 255.0.0.0 10.0.0.2 3
  ```

  PIX Firewall converts the command statement to the following:

  ```
  route inside 10.0.0.0 255.0.0.0 10.0.0.2 3 OTHER
  ```

Examples

Specify one default **route** command statement for the outside interface, which in this example is for the router on the outside interface that has an IP address of 209.165.201.1:

```
route outside 0 0 209.165.201.1 1
```

For static routes, if two networks, 10.1.2.0 and 10.1.3.0 connect via a hub to the dmz1 interface router at 10.1.1.4, add these static **route** command statements to provide access to the networks:

```
route dmz1 10.1.2.0 255.0.0.0 10.1.1.4 1
route dmz1 10.1.3.0 255.0.0.0 10.1.1.4 1
```

route-map

Defines the conditions for redistributing routes from one routing protocol into another. (Used in configuring OSPF routing on the firewall.) OSPF routing is not supported on the PIX 501.

[**no**] **route-map** *map_tag* [**permit** | **deny**] [*seq_num*]

show route-map [*map_tag*]

Subcommands to the **route-map** command:

[**no**] **match** [**interface** *interface_name* | **metric** *metric_value* | **ip address** *acl_id* | **route-type** {**local** | **internal** | [**external** [**type-1** | **type-2**]]}| **nssa-external** [**type-1** | **type-2**] | **ip next-hop** *acl_id* | **ip route-source** *acl_id*]

[**no**] **set metric** *value*

[**no**] **set metric-type** {**type-1** | **type-2** | **internal** | **external**}

[**no**] **set ip next-hop** *ip-address* [*ip-address*]

Syntax Description

acl_id	The name of an ACL. The **match ip next-hop** and **match ip route-source** commands can accept more than one *acl_id*. That is, they accept *acl_id* [*...acl_id*].
deny	If the match criteria are met for the route map and the **deny** option is specified, the route is not redistributed.
external	The OSPF metric routes external to a specified autonomous system.
interface_name	The name of the interface.
internal	Routes that are internal to a specified autonomous system.
ip next-hop *ip-address* [*ip-address*]	Indicates where to output packets that pass a match clause of the route map.
ip route-source	Redistributes routes that have been advertised by routers and access servers at the address specified by the *acl_id*.
local	Specifies a preference value for the autonomous system path.
map_tag	The text for the route map tag, meant to define a meaningful name for the route map, up to 58 characters in length. Multiple route maps may share the same map tag name.
metric_value	A metric value, from 0 to 2147483647.
nssa-external [**type-1** | **type-2**]	The OSPF metric type for routes that are external to a not-so-stubby area (NSSA), either type 1 or 2. The default is type 2.
permit	If the match criteria are met for this route map, and the **permit** option is specified, the route is redistributed as controlled by the set actions. If the match criteria are not met, and the **permit** keyword is specified, the next route map with the same *map_tag* is tested. If a route passes none of the match criteria for the set of route maps sharing the same name, it is not redistributed by that set. The **permit** option is the default.
seq_num	If there are any route maps with the same *map_tag*, then you must also specify a *seq_num* for the route-maps to differentiate between them. The *seq_num* can be any number from 0 to 65535. Otherwise, no *seq_num* needs to be specified. A default value of 10 is assigned to the first route map if no *seq_num* is specified. If given in the **no route-map** *map_tag seq_num* command, *seq_num* is the route map to be deleted.
type-1 | type-2	The OSPF metric routes external to a specified autonomous system, either type 1 or 2. The default is type 2.

Defaults

The **permit** option is the default for the **route-map** command.

Command Modes

The **route-map** command is available in configuration mode.

The **show route-map** command is available in privileged mode.

Usage Guidelines

To define the conditions for redistributing routes from one routing protocol into another, use the **route-map** *map_tag* command and the **match** and **set route-map** configuration commands. To delete an entry, use the **no route-map** *map_tag* command.

set metric *value*

To set the metric value for a routing protocol, use the **set metric** *value* subcommand. To return to the default metric value, use the **no set metric** *value* subcommand. In this context, the *value* is an integer from -2147483647 to 2147483647.

set metric-type {**type-1 | type-2**}

To set the metric type for the destination routing protocol, use the **set metric-type** {**type-1 | type-2**} subcommand. To return to the default, use the **no set metric-type** {**type-1 | type-2**} subcommand.

set ip next-hop *ip-address*

To indicate where to output packets that pass a match clause of a route map, use the **set ip next-hop** *ip_address* subcommand. To delete an entry, use the **no set ip next-hop** *ip-address* subcommand. In this context, *ip_address* is the IP address of the next hop to which to output packets. It must be the address of an adjacent router.

Examples

The following example show how to configure a route map for use in OSPF routing:

```
pixfirewall(config)# route-map maptag1 permit 8
pixfirewall(config-route-map)# set metric 5
pixfirewall(config-route-map)# match metric 5
pixfirewall(config-route-map)# set metric-type type-2
pixfirewall(config-route-map)# show route-map
route-map maptag1 permit 8
  set metric 5
  set metric-type type-2
  match metric 5
pixfirewall(config-route-map)# exit
pixfirewall(config)#
```

Related Commands

prefix-list	Configures a prefix list to be used for OSPF routing.
router ospf	Configures global parameters for the OSPF routing processes on the firewall, and enables or disables OSPF routing through the firewall.
routing interface	Configures interface-specific OSPF routing parameters.

router ospf

Configures global parameters for the OSPF routing processes on the firewall, and enables or disables OSPF routing through the firewall. (Use the **routing interface** command for interface-specific OSPF configration.) OSPF routing is not supported on the PIX 501.

[**no**] **router ospf** *pid*

show router ospf *pid*

Subcommands to the **router ospf** command:

[**no**] **area** *area_id*

[**no**] **area** *area_id* **authentication** [**message-digest**]

[**no**] **area** *area_id* **default-cost** *cost*

[**no**] **area** *area_id* **filter-list prefix** {*prefix_list_name* **in | out**}

[**no**] **area** *area_id* **nssa** [**no-redistribution**] [**default-information-originate** [**metric-type 1 | 2**] [**metric** *metric_value*]]

[**no**] **area** *area_id* **range** *ip_address netmask* [**advertise | not-advertise**]

area *area_id* **stub** [**no-summary**]

[**no**] **area** *area_id* **virtual-link** *router_id* [**authentication** [**message-digest | null**]] [**hello-interval** *seconds*] [**retransmit-interval** *seconds*] [**transmit-delay** *seconds*] [**dead-interval** *seconds*] [**authentication-key** *password*] [**message-digest-key** *id* **md5** *password*]

[**no**] **compatible rfc1583**

default-information originate [**always**] [**metric** *metric_value*] [**metric-type** {**1 | 2**}] [**route-map** *map_name*]

[**no**] **distance ospf** [**intra-area** *d1*][**inter-area** *d2*][**external** *d3*]

[**no**] **ignore lsa mospf**

[**no**] **log-adj-changes** [**detail**]

[**no**] **network** *prefix ip_address netmask* **area** *area_id*

[**no**] **redistribute** {**static | connected**} [**metric** *metric_value*] [**metric-type** *metric_type*] [**route-map** *map_name*] [**tag** *tag_value*] [**subnets**]

[**no**] **redistribute ospf** *pid* [**match** {**internal | external** [**1|2**] **| nssa-external** [**1|2**]}] [**metric** *metric_value* **] [metric-type** *metric_type*] **[route-map** *map_name*] **[tag** *tag_value*] **[subnets**]

[**no**] **router-id** *ip_address*

[**no**] **summary-address** *addr netmask* [**not-advertise**] [**tag** *tag_value*]

[**no**] **timers** {**spf** *spf_delay spf_holdtime* **| lsa-group-pacing** *seconds*}

Syntax Description

addr	The value of the summary address designated for a range of addresses.
advertise	Sets the address range status to advertise and generates a Type 3 summary link-state advertisements (LSA).
area *area_id*	Configures a regular OSPF area.
area_id	For all contexts, *area_id* can be specified as either a decimal value or as an IP address. The ID of the area that is to be associated with the OSPF address range. If you intend to associate areas with IP subnets, you can specify a subnet address as the *area_id*. When used in the context of authentication, *area_id* is the identifier of the area on which authentication is to be enabled. When using a cost context, *area_id* is the identifier for the stub or NSSA. When used in the context of a prefix list, *area_id* is the identifier of the area on which filtering is configured. When used in a stub area or not-so-stubby area (NSSA) context, *area_id* is the identifier for the stub or NSSA area. When used in the context of an area range, *area_id* is the identifier of the area at whose boundary to summarize routes.
authentication	(Optional) Specifies the authentication type.
compatible	Runs OSPF in RFC 1583 compatible mode.
cost	The cost for the default summary route used for a stub or NSSA, from 0 to 65535. The default value for *cost* is 1.
d1, *d2*, and *d3*	The distance for different area route types. The default for *d1*, *d2*, and *d3* is 110.
default-information	Distributes a default route according to the parameters specified.
default-information -originate	Used to generate a Type 7 default in the NSSA area. This keyword only takes effect on an NSSA ABR or an NSSA Autonomous System Boundary Router (ASBR).
distance	Configures administrative distances for the OSPF process.
external	Sets the distance for routes from other routing domains, learned by redistribution.
external 1 | 2	The OSPF metric routes external to a specified autonomous system, either type 1 or 2. The default is type 2.
ignore	Supresses syslog for receipt of type 6 Multicast OSPF LSAs.
in	Applies the configured prefix list to prefixes advertised inbound to the specified area.
inter-area	Sets the distance for all routes from one area to another area.
internal	Routes that are internal to a specified autonomous system.
ip_address	The router ID in IP address format.
log-adj-changes	Logs OSPF adjacency changes.
lsa-group-pacing *seconds*	The interval at which OSPF link-state advertisements (LSAs) are collected into a group and refreshed, checksummed, or aged, from 10 to 1800 seconds. The default value is 240 seconds.
map_name	The name of the route map to apply.

message-digest	(Optional) Enables Message Digest 5 (MD5) authentication on the area specified by the *area_id*.
metric *metric_value*	Specifies the OSPF default metric value, from 0 to 16777214.
netmask	An IP address mask, or IP subnet mask used for a summary route.
network	Adds/removes interfaces to/from the OSPF routing process.
no-redistribution	When the OSPF router is an NSSA Area Border Router (ABR) and you want the **redistribute** command to import routes only into the normal areas, and not into the NSSA area, use this option.
no-summary	Prevents an Area Border Router (ABR) from sending summary link-state advertisements into the stub area.
not-advertise	Sets the address range status to DoNotAdvertise. The Type 3 summary LSA is suppressed, and the component networks remain hidden from other networks. In the **summary-address** command, **not-advertise** suppresses routes that match the specified prefix/mask pair.
nssa-external 1 | 2	The OSPF metric type for routes that are external to a not-so-stubby area (NSSA), either type 1 or 2. The default is type 2.
null	(Optional) Specifies that no authentication is used. Overrides password or message digest authentication if configured for the OSPF area.
out	Applies the configured prefix list to prefixes advertised outbound from the specified area.
pid	Internally used identification parameter for an OSPF routing process. You assign it locally on the firewall, and it can be from 1 to 65535. A unique value must be assigned for each OSPF routing process. PIX Firewall software Version 6.3 supports a maximum of two (2) OSPF processes.
prefix	Indicates that a prefix list is used. (Prefix lists are configured with the **prefix-list** command.)
prefix	An IP address.
prefix_list_name	Name of a prefix list.
redistribute	Configures redistribution between OSPF processes according to the parameters specified.
router-id	Configures the router ID for an OSPF process.
seconds	The interval at which OSPF link-state advertisements (LSAs) are collected into a group and refreshed, checksummed, or aged, from 10 to 1800 seconds. The default is 240 seconds.
spf_delay	The delay time between when OSPF receives a topology change and when it starts a shortest path first (SPF) calculation in seconds, from 0 to 65535. The default is 5 seconds.
spf_holdtime	The hold time between two consecutive SPF calculations in seconds, from 0 to 65535. The default is 10 seconds.
stub	An OSPF area that carries a default route and intra- and inter-area routes but does not carry external routes. Virtual links cannot be configured across a stub area, and they cannot contain an autonomous system boundary router (ASBR).
subnets	(Optional) For redistributing routes into OSPF, scopes the redistribution for the specified protocol.
summary-address	Configures the summary address for OSPF redistribution.

tag_value	The value to match (for controlling redistribution with route maps).
timers	Configures timers for the OSPF process.

Defaults

The default is for OSPF routing to be disabled on the firewall.

The default value for *cost* is 1.

The default authentication type for an area is **0**, which means no authentication.

By default, OSPF routing through the firewall is compatible with RFC 1583.

The default for the **area** *area_id* **range** *ip_address netmask* [**advertise | not-advertise**] command is **advertise**.

The default for *d1*, *d2*, and *d3* in the **distance ospf** [**intra-area** *d1*][**inter-area** *d2*][**external** *d3*] subcommand is is 110.

By default, the **log-adj-changes** subcommand is enabled.

The default for *spf_delay* is 5 seconds, and the default for *spf_holdtime* is 10 seconds.

The default for the **timers lsa-group-pacing** *seconds* subcommand is 240 seconds.

No area is defined by default for the **area** *area_id* **nssa no-redistribution** or **area** *area_id* **default-information-originate** subcommands.

Command Modes

The **router ospf** command is available in configuration mode.

The **show router ospf** command is available in privileged mode.

Usage Guidelines

The **router ospf** command is the global configuration command for OSPF routing processes running on the firewall. This is the main command for all of the OSPF configuration commands.

Note Open Shortest Path First (OSPF) is used instead of Routing Information Protocol (RIP). Do not attempt to configure the firewall for both OSPF and RIP simeltaneously.

When using the **no** form of a **router ospf** command, optional arguments need not be specified unless they provide necessary information. The **no router ospf** command terminates the OSPF routing process specified by its *pid*.

PIX Firewall software Version 6.3 supports a maximum of two (2) OSPF processes.

The **show ospf** command displays the configured **router ospf** subcommands.

OSPF areas

The **area** *area_id* subcommand creates a regular OSPF area. The **no area** *area_id* command removes the OSPF area, whether it is regular, stubby, or not-so-stubby.

area *area_id* authentication message-digest

The default authentication type for an area is **0**, which means no authentication. To enable authentication for an OSPF area, use the **area** *area_id* **authentication message-digest** subcommand. To remove an authentication configuration from an area, use the **no area** *area_id* **authentication message-digest** subcommand.

area *area_id* **default-cost** *cost*

To specify a cost for the default summary route sent into a stub or not-so-stubby area (NSSA), use the **area** *area_id* **default-cost** *cost* subcommand. To remove the assigned default route cost, use the **no area** *area_id* **default-cost** subcommand. The default value for *cost* is 1.

area *area_id* **filter-list prefix** *prefix_list_name* [**in | out**]

To filter prefixes advertised in type 3 link-state advertisements (LSAs) between Open Shortest Path First (OSPF) areas of an Area Border Router (ABR), use the **area** *area_id* **filter-list prefix** *prefix_list_name* [**in | out**] subcommand. To change or cancel the filter, use the no **area** *area_id* **filter-list prefix** *prefix_list_name* [**in | out**] subcommand.

area *area_id* **nssa** [**no-redistribution**] [**default-information-originate** [**metric-type 1 | 2**] [**metric** *metric_value*]]

Routes that originate from other routing protocols (or different OSPF processes) and that are injected into OSPF through redistribution are called external routes. There are two forms of external metrics: type 1 and type 2. These routes are represented by `O E2` (for type 2) or `O E1` (for type 1) in the IP routing table, and they are examined after the firewall is done building its internal routing table. After they are examined, they are flooded throughout the autonomous systems (AS), unaltered. (Autonomous systems are a collection of networks, subdivided by areas, under a common administration sharing a common routing strategy.)

OSPF type 1 metrics result in routes adding the internal OSPF metric to the external route metric; they are also expressed in the same terms as an OSPF link-state metric. The internal OSPF metric is the total cost of reaching the external destination, including whatever internal OSPF network costs are incurred to get there. (These costs are calculated by the device wanting to reach the external route.) Because it is calculated this way, the OSPF type 1 metric is generally preferred.

OSPF type 2 metrics do not add the internal OSPF metric to the cost of external routes and are the default type used by OSPF. The use of OSPF type 2 metrics assumes that you are routing between autonomous systems (AS); therefore, the cost is considered greater than any internal metrics. This eliminates the need to add the internal OSPF metrics.

To configure an area as a not-so-stubby area (NSSA), use the **area** *area_id* **nssa** [**no-redistribution**] [**default-information-originate** [**metric-type 1 | 2**] [**metric** *metric_value*]] subcommand. To remove the entire NSSA configuration, use the **no area** *area_id* **nssa** subcommand. To remove a single NSSA configuration option, specify the option in the **no** subcommand. For example, to remove the **no-redistribution** option, use the **no area** *area_id* **nssa no-redistribution** command. By default, no NSSA is defined.

area *area_id* **range** *address netmask* [**advertise | not-advertise**]

To consolidate and summarize routes at an area boundary, use the **area** *area_id* **range** *address netmask* [**advertise | not-advertise**] subcommand. To disable this function, use the **no area** *area_id* **range** *ip_address netmask* subcommand. The **no area** *area_id* **range** *ip_address netmask* **not-advertise** subcommand removes only the **not-advertise** option.

area *area_id* **stub** [**no-summary**]

To define an area as a stub area, use the **area** *area_id* **stub** [**no-summary**] subcommand. To remove the stub area function, use the no **area** *area_id* **stub** [**no-summary**] subcommand. When **area** *area_id* **stub no-summary** is configured, you must use **no area** *area_id* **stub no-summary** to remove the no summary option. The default is for no stub areas to be defined.

[**no**] **area** *area_id* **virtual-link** *router_id* [**hello-interval** *seconds*] [**retransmit-interval** *seconds*] [**transmit-delay** *seconds*] [**dead-interval** *seconds*] [**authentication-key** *password*] [**message-digest-key** *id* **md5**] *password*

To define an OSPF virtual link, use the **area** *area_id* **virtual-link** *router-id* subcommand with the optional parameters. To remove a virtual link, use the **no area** *area_id* **virtual-link** *router_id* subcommand.

compatible rfc1583

To restore the method used to calculate summary route costs per RFC 1583, use the **compatible rfc1583** subcommand. To disable RFC 1583 compatibility, use the **no compatible rfc1583** subcommand.

By default, OSPF routing through the firewall is compatible with RFC 1583. The **compatible rfc1583** subcommand is displayed in the configuration only if disabled by the **no compatible rfc1583** subcommand, and then as "`no compatible rfc1583`".

distance ospf [intra-area *d1*][inter-area *d2*][external *d3*]

To define OSPF route administrative distances based on route type, use the **distance ospf** [**intra-area** *d1*][**inter-area** *d2*][**external** *d3*] subcommand. To restore the default value, use the **no distance ospf** subcommand. The default for *d1*, *d2*, and *d3* is 110.

ignore lsa mospf

To suppress the sending of syslog messages when the router receives link-state advertisement (LSA) for Type 6 Multicast OSPF (MOSPF) packets, which are unsupported, use the **ignore lsa mospf** subcommand. To restore the sending of these syslog messages, use the **no ignore lsa mospf** subcommand.

log-adj-changes

To configure the router to send a syslog message when an OSPF neighbor goes up or down, use the **log-adj-changes** [**detail**] subcommand. To turn off this function, use the **no log-adj-changes** subcommand. The **detail** option sends a syslog message for each state change, not just when a neighbor goes up or down.

By default, the **log-adj-changes** subcommand is enabled, but the **log-adj-changes** subcommand is only displayed in the OSPF configuration when the **detail** option is specified or when it has been disabled.

network *prefix ip_address netmask* area *area_id*

To define the interfaces on which OSPF runs and the area ID for those interfaces, use the **network** *prefix ip_address netmask* **area** *area_id* subcommand. To disable OSPF routing for the interfaces defined with the *prefix ip_address netmask* pair, use the **no network** *prefix ip_address netmask* **area** *area_id* subcommand.

summary-address *addr netmask*

To create aggregate addresses for OSPF, use the **summary-address** *addr netmask* [**not-advertise**] [**tag** *tag*] subcommand. To restore the default, use the no **summary-address** *addr netmask* subcommand. The *addr* value is the summary address designated for a range of addresses, and *netmask* is the IP subnet mask used for the summary route.

router-id

To use a fixed router ID, use the **router-id** *address* subcommand. To reset OSPF to use the previous OSPF router ID behavior, use the **no router-id** subcommand.

Note If the highest-level IP address on the firewall is a private address, then this address is sent in hello packets and database definitions (DBDs). To prevent this, set the **router-id** *ip_ddress* to a global address.

timers {**spf** *spf_delay spf_holdtime* | **lsa-group-pacing** *seconds*}

To configure the delay time between when OSPF receives a topology change and when it starts a shortest path first (SPF) calculation, and the hold time between two consecutive SPF calculations, use the **timers spf** *spf_delay spf_holdtime* subcommand. To return to the default timer values, use the **no timers spf** subcommand.

To change the interval at which OSPF link-state advertisements (LSAs) are collected into a group and refreshed, checksummed, or aged, use the **timers lsa-group-pacing** *seconds* subcommand. To restore the default value, use the no **timers lsa-group-pacing** *seconds* subcommand. The default for *seconds* is 240.

Examples

To enter subcommand mode on the outside interface of the firewall (needed to configure OSPF routing), enter the following command:

```
pixfirewall(config)# router ospf 5
pixfirewall(config-router)#
```

When in the routing subcommand mode, the command prompt appears as "**(config-router)#**".

Related Commands

prefix-list	Configures a prefix list to be used for OSPF routing.
route-map	Creates a route map for redistributing routes from one routing protocol to another. Used in configuring OSPF routing on the firewall.
routing interface	Configures interface-specific OSPF routing parameters.

routing interface

Configures interface-specific OSPF routing parameters. This command is the main command for all OSPF interface submode commands. (Use the **router ospf** command to configure global parameters and to enable OSPF routing through the firewall.) OSPF routing is not supported on the PIX 501.

[**no**] **routing interface** *interface_name*

Subcommands to the **routing interface** command:

[**no**] **ospf authentication** [**message-digest** | **null**]

[**no**] **ospf authentication-key** *password*

[**no**] **ospf cost** *interface_cost*

[**no**] **ospf database-filter all out**

[**no**] **ospf dead-interval** *seconds*

[**no**] **ospf hello-interval** *seconds*

[**no**] **ospf message-digest-key** *key-id* **md5** *key*

[**no**] **ospf mtu-ignore**

[**no**] **ospf priority** *number*

[**no**] **ospf retransmit-interval** *seconds*

[**no**] **ospf transmit-delay** *seconds*

Syntax Description

authentication-key *password*	Assigns an OSPF authentication password for use by neighboring routing devices. This can be any continuous string of keyboard characters, except for whitespace characters such as tabs or spaces, up to 8 bytes in length.
database-filter all out	Filters out outgoing link-state advertisements (LSAs) to an OSPF interface.
dead-interval *seconds*	Sets the interval before declaring a neighboring routing device is down if no hello packets are received, from 1 to 65535 seconds. This value must be the same for all nodes on the network. The default is four times the interval set by the **ospf hello-interval** command.
hello-interval *seconds*	Specifies the interval between hello packets sent on the interface, from 1 to 65535 seconds. The default is 10 seconds.
interface_cost	The cost (a link-state metric) of sending a packet through an interface. This is an unsigned integer value from 0 to 65535. **0** represents a network that is directly connected to the interface, and the higher the interface bandwidth, the lower the associated cost to send packets across that interface. In other words, a large cost value represents a low bandwidth interface and a small cost value represents a high bandwidth interface. The OSPF interface default cost on the firewall is **10**. This default differs from Cisco IOS software, where the default cost is **1** for fast Ethernet (FE) and Gigabit Ethernet (GE) and **10** for 10BaseT. This is important to take into account if you are using Equal Cost Multi-Path (ECMP) in your network.
interface_name	The name of the interface to configure.
key_id	A numerical ID number, from 1 to 255, for the authentication key.
md5 *key*	An alphanumeric password of up to 16 bytes. However, whitespaces characters such as a tab or space are not supported.
message-digest	Specifies to use OSPF message digest authentication.
message-digest-key	Enables Message Digest 5 (MD5) authentication. (MD5 verifies the integrity of the communication, authenticates the origin, and checks for timeliness.)
null	Specifies to not use OSPF authentication. This overrides password or message digest authentication (if configured) for an OSPF area.
ospf	Keyword for configuring interface-specific OSPF parameters.
priority *number*	A positive integer from 0 to 255 that specifies the priority of the router. The default is 1.
retransmit-interval *seconds*	Specifies the time between link-state advertisement (LSA) retransmissions for adjacent routers belonging to the interface, from 1 to 65535 seconds. The default is 5 seconds.
transmit-delay *seconds*	Sets the estimated time required to send a link-state update packet on the interface, from 1 to 65535 seconds. The default is 1 second.

Defaults

By default, OSPF routing is disabled on the firewall interfaces.

By default, the **mtu-ignore** subcommand is enabled.

The default value for the **ospf authentication** [**message-digest | null**] subcommand is **null**, which means no area authentication.

The default value for the **ospf dead-interval** subcommand is four times the interval set by the **ospf hello-interval** command.

The default value for the **ospf hello-interval** subcommand is 10 seconds.

The default value for the **ospf retransmit-interval** subcommand is 5 seconds.

The default value for the **ospf transmit-delay** subcommand is 1 second.

Command Modes

The **routing** command is available in configuration mode.

The **show routing** command is available in privileged mode.

Usage Guidelines

The **routing interface** *interface_name* command is the main command for all interface-specific OSPF interface mode commands. Enter this command with the name of the firewall interface (*interface_name*) that you want to configure, and then proceed with interface-specific configuration through the **routing interface** subcommands. You do not need to specify optional arguments in the **no** forms of the **routing interface** subcommands (unless they provide necessary information).

The **no routing interface** *interface_name* command removes the routing configuration for the interface specified only.

The **clear routing** command resets the interface-specific routing configuration to its defaults and removes the interface-specific routing configuration. However, this command does not remove any OSPF data structures that have been defined.

The **clear ospf** [*pid*] {**process | counters | neighbor** [**neighbor-intf**] [**neighbr-id**]} command resets the OSPF routing process ID, counters, neighbor interface router designation, or neighbor router ID, depending on the option selected. This command does not remove any configuration. Use the no form of the **router ospf** or **routing interface** command to remove the OSPF configuration.

The **show routing interface** *interface_name* command displays the configuration for the interface specified.

ospf authentication

To specify the authentication type for an interface, use the **ospf authentication** [**message-digest | null**] subcommand. To remove the authentication type for an interface, use the **no ospf authentication** [**message-digest | null**] subcommand. The default area authentication is **null**, which means no authentication.

ospf authentication-key

To assign a password to be used by neighboring routers that are using the OSPF simple password authentication, use the **ospf authentication-key** *password* subcommand. The variable *password* can be any continuous string of characters that can be entered from the keyboard, up to 8 bytes in length.

To remove a previously assigned OSPF password, use the **no ospf authentication-key** subcommand.

ospf cost

To explicitly specify the cost of sending a packet on an interface, use the **ospf cost** *interface_cost* subcommand. The *interface_cost* parameter is an unsigned integer value from 0 to 255, expressed as the link-state metric.

To reset the path cost to the default value, use the **no ospf cost** subcommand.

ospf database-filter all out

To filter outgoing link-state advertisements (LSAs) to an OSPF interface, use the **ospf database-filter** subcommand. To restore the forwarding of LSAs to the interface, use the **no ospf database-filter all out** subcommand.

ospf dead-interval

To set the dead interval before neighbors declare the router down (the length of time during which no hello packets are seen), use the **ospf dead-interval** *seconds* subcommand. The variable *seconds* specifies the dead interval and must be the same for all nodes on the network. The default for *seconds* is four times the interval set by the **ospf hello-interval** command, from 1 to 65535. To return to the default interval value, use the **no ospf dead-interval** subcommand.

ospf hello-interval

To specify the interval between hello packets that the firewall sends on the interface, use the **ospf hello-interval** *seconds* subcommand. To return to the default interval, use the **no ospf hello-interval** subcommand. The default is 10 seconds, with a range from 1 to 65535.

ospf mtu-ignore

The **ospf mtu-ignore** subcommand disables OSPF MTU mismatch detection on receiving DBD packets and is enbled by default.

ospf message-digest-key *key_id* md5 *key*

To enable OSPF Message Digest 5 (MD5) authentication, use the **ospf message-digest-key** *key_id* **md5** *key* subcommand. To remove an old MD5 key, use the **no ospf message-digest-key** *key_id* **md5** *key* subcommand. The *key_id* variable is a numerical identifier, from 1 to 255, for the authentication key, and the *key* variable is an alphanumeric password of up to 16 bytes.

ospf priority

To set the router priority, which helps determine the designated router for this network, use the **ospf priority** *number* subcommand. To return to the default value, use the **no ospf priority** *number* subcommand.

ospf retransmit-interval

To specify the time between link-state advertisement (LSA) retransmissions for adjacencies belonging to the interface, use the **ospf retransmit-interval** *seconds* subcommand. To return to the default value, use the **no ospf retransmit-interval** subcommand. The default value is 5 seconds, with a range from 1 to 65535.

ospf transmit-delay

To set the estimated time required to send a link-state update packet on the interface, use the **ospf transmit-delay** *seconds* subcommand. To return to the default value, use the **no ospf transmit-delay** subcommand. The default value is 1 second, with a range from 1 to 65535.

Examples

To enter subcommand mode on the outside interface of the firewall (needed to configure OSPF routing), enter the following command:

```
pixfirewall(config)# routing interface outside
pixdocipsec1(config-routing)#
```

When in the routing subcommand mode, the command prompt appears as "**(config-routing)#**".

The following example shows the configuration for two concurrently running OSPF processes, with the IDs 5 and 12, on the outside interface of the firewall:

```
pixfirewall(config)# routing interface
pixfirewall(config-routing)# show ospf

 Routing Process "ospf 5" with ID 127.0.0.1 and Domain ID 0.0.0.5
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
 Number of external LSA 0. Checksum Sum 0x     0
 Number of opaque AS LSA 0. Checksum Sum 0x     0
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 0. 0 normal 0 stub 0 nssa
 External flood list length 0

 Routing Process "ospf 12" with ID 172.23.59.232 and Domain ID 0.0.0.12
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
 Number of external LSA 0. Checksum Sum 0x     0
 Number of opaque AS LSA 0. Checksum Sum 0x     0
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 0. 0 normal 0 stub 0 nssa
 External flood list length 0
```

The following example changes the retransmit interval to 15 seconds:

```
pixdocipsec1(config-routing)# ospf retransmit-interval 15
```

Related Commands

prefix-list	Configures a prefix list to be used for OSPF routing.
route-map	Creates a route map for redistributing routes from one routing protocol to another. Used in configuring OSPF routing on the firewall.
router ospf	Configures global parameters for the OSPF routing processes on the firewall, and enables or disables OSPF routing through the firewall.

S Commands

service

Enable system services.

[no] service {resetinbound | resetoutside}

clear service

show service

Syntax Description

resetinbound	Send a reset to a denied inbound TCP packet.
resetoutside	Send a reset to a denied TCP packet to outside interface.

Command Modes

Configuration mode.

Usage Guidelines

The **service** command works with all inbound TCP connections to statics whose access lists or uauth (user authorization) do not allow inbound. One use is for resetting IDENT connections. If an inbound TCP connection is attempted and denied, you can use the **service resetinbound** command to return an RST (reset flag in the TCP header) to the source. Without the option, the PIX Firewall drops the packet without returning an RST.

For use with IDENT, the PIX Firewall sends a TCP RST to the host connecting inbound and stops the incoming IDENT process so that email outbound can be transmitted without having to wait for IDENT to time out. In this case, the PIX Firewall sends a syslog message stating that the incoming connection was a denied connection. Without **service resetinbound**, the PIX Firewall drops packets that are denied and generates a syslog message stating that the SYN was a denied connection. However, outside hosts keep retransmitting the SYN until the IDENT times out.

When an IDENT connection is timing out, you will notice that connections slow down. Perform a trace to determine that IDENT is causing the delay and then invoke the **service** command.

The **service resetinbound** command provides a safer way to handle an IDENT connection through the PIX Firewall. Ranked in order of security from most secure to less secure are these methods for handling IDENT connections:

1. Use the **service resetinbound** command.
2. Use the **established** command with the **permitto tcp 113** options.
3. Enter **static** and **access-list** command statements to open TCP port 113.

When using the **aaa** command, if the first attempt at authorization fails and a second attempt causes a timeout, use the **service resetinbound** command to reset the client that failed the authorization so that it will not retransmit any connections. An example authorization timeout message in Telnet follows:

```
Unable to connect to remote host: Connection timed out
```

Examples

The following example shows use of the **service resetinbound** command:

```
service resetinbound
show service
service resetinbound
```

If you use the **resetoutside** command, the PIX Firewall actively resets denied TCP packets that terminate at the PIX Firewall unit's least-secure interface. By default, these packets are silently discarded. The **resetoutside** option is highly recommended with dynamic or static interface Port Address Translation (PAT). The static interface PAT is available with PIX Firewall Version 6.0 and higher. This option allows the PIX Firewall to quickly terminate the identity request (IDENT) from an external SMTP or FTP server. Actively resetting these connections avoids the thirty-second time-out delay.

If you wish to remove **service** command statements from the configuration, use the **clear service** command.

session enable

The **session enable** command is a deprecated command.

setup

The **setup** command prompts you to enter the information needed to use the Cisco PIX Device Manager (PDM) with a new PIX Firewall.

setup

Syntax Description

setup	Asks for the information needed to start using a new PIX Firewall unit if no configuration is found in the Flash memory.

Command Modes

Configuration mode.

Usage Guidelines

The PIX Firewall requires some pre-configuration before PDM can connect to it. (The setup dialog automatically appears at boot time if there is no configuration in the Flash memory.) Once you enter the **setup** command, you will be asked for the setup information in Table 8-1.

Table 8-1 PIX Firewall Setup Information

Prompt	Description
`Enable password:`	Specify an enable password for this PIX Firewall. (The password must be at least three characters long.)
`Clock (UTC)`	Set the PIX Firewall clock to Universal Coordinated Time (also known as Greenwich Mean Time).
`Year [`*`system year`*`]:`	Specify current year, or default to the year stored in the host computer.
`Month [`*`system month`*`]:`	Specify current month, or default to the month stored in the host computer.
`Day [`*`system day`*`]:`	Specify current day, or default to the day stored in the host computer.
`Time [`*`system time`*`]`	Specify current time in *hh:mm:ss* format, or default to the time stored in the host computer.
`Inside IP address:`	Network interface IP address of the PIX Firewall.
`Inside network mask:`	A network mask that applies to the inside IP address must be a valid mask such as 255.0.0.0, 255.255.0.0, or 255.255.x.x, etc. Use **0.0.0.0** to specify a default route. The **0.0.0.0** netmask can be abbreviated as **0**.
`Host name:`	The host name you want to display in the PIX Firewall command line prompt.
`Domain name:`	The DNS domain name of the network on which the PIX Firewall runs, for example *example.com*.
`IP address of host running PIX Device Manager:`	IP address on which PDM connects to the PIX Firewall.
`Use this configuration and write to flash?`	Store the new configuration to Flash memory. Same as the **write memory** command. If the answer is **yes**, the inside interface will be enabled and the requested configuration will be written to Flash memory. If the user answers anything else, the setup dialog repeats using the values already entered as the defaults for the questions.

The host and domain names are used to generate the default certificate for the SSL connection. The interface type is determined by the hardware.

Examples

The following example shows how to complete the **setup** command prompts.

```
router (config)# setup
Pre-configure PIX Firewall now through interactive prompts [yes]? y
Enable Password [<use current password>]: ciscopix
Clock (UTC)
  Year [2001]: 2001
  Month [Aug]: Sep
  Day [27]: 12
  Time [22:47:37]: <Enter>
Inside IP address: 192.168.1.1
Inside network mask: 255.255.255.0
Host name: accounting_pix
```

```
Domain name: example.com
IP address of host running PIX Device Manager: 192.168.1.2

The following configuration will be used:
Enable Password: ciscopix
Clock (UTC): 22:47:37 Sep 12 2001
Inside IP address: ...192.168.1.1
Inside network mask: ...255.255.255.0
Host name: ...accounting_pix
Domain name: ...example.com
IP address of host running PIX Device Manager: ...192.168.1.2

Use this configuration and write to flash? y
```

Related Commands

pdm	Configures PIX Device Manager (PDM).

show

View command information.

show *command_keywords* [| {**include** | **exclude** | **begin** | **grep** [**-v**]} *regexp*]

show ?

Syntax Description

command_key words	Any argument or list of arguments that specifies the information to display. Most commands have a **show** command form where the command name is used as **show** argument. For example, the **global** command has an associated **show global** command.
|	The UNIX pipe symbol, "|". This character represents piping output to the filter. When "|" is present, a filtering option and a regular expression must also be present. (Only the first "|" is a pipe character in the syntax.)
include	Includes all output lines that match the specified regular expression.
exclude	Excludes all output lines that match the specified regular expression.
grep	Displays all output lines that match the specified regular expression. **grep** is equivalent to **include** and **grep -v** is equivalent to **exclude**.
begin	Displays all output lines starting from the line that matches the specified regular expression.
regexp	A Cisco IOS software style regular expression. Do not enclose in quotes or double-quotes. Additionally, trailing white spaces (between keywords) are taken as part of the regular expression.

Command Modes

All modes.

Usage Guidelines

The **show** *command_keywords* [| {**include | exclude | begin | grep**} *regexp*] command runs the show command options specified. See individual commands for their show options. (Only the first "|" is a pipe character in this syntax.) The CLI syntax and semantics of the **show** output filtering options are the same as in Cisco IOS software, and are available through console, Telnet, or SSH sessions.

The **show ?** command displays a list of all commands available on the PIX Firewall.

Explanations for the specific **show** commands are documented with the corresponding command. For example, the **show arp** command description is included with the **arp** command.

Examples

The following example illustrates how to use a show command output filter option, where the "|" is the UNIX pipe symbol:

```
pixfirewall(config)# show config | grep access-list
access-list 101 permit tcp any host 10.1.1.3 eq www
access-list 101 permit tcp any host 10.1.1.3 eq smtp
```

The following is sample output from the **show ?** command:

```
pixfirewall(config)# show ?

At the end of show <command>, use the pipe character '|' followed by:
begin|include|exclude|grep [-v] <regular_exp>, to filter show output.

aaa             Enable, disable, or view TACACS+, RADIUS or LOCAL
                user authentication, authorization and accounting

aaa-server      Define AAA Server group
access-group    Bind an access-list to an interface to filter inbound traffic
access-list     Add an access list
activation-key  Modify activation-key.
age             This command is deprecated. See ipsec, isakmp, map, ca commands
alias           Administer overlapping addresses with dual NAT.
apply           Apply outbound lists to source or destination IP addresses
arp             Change or view arp table, set arp timeout value and view statiss
auth-prompt     Customize authentication challenge, reject or acceptance prompt
auto-update     Configure auto update support
banner          Configure login/session banners
blocks          Show system buffer utilization
ca              CEP (Certificate Enrollment Protocol)
                Create and enroll RSA key pairs into a PKI (Public Key Infrastr.
capture         Capture inbound and outbound packets on one or more interfaces
checksum        View configuration information cryptochecksum
chunkstat       Display chunk stats
clock           Show and set the date and time of PIX
conduit         Add conduit access to higher security level network or ICMP
configure       Configure from terminal, floppy, memory, network, or
                factory-default.  The configuration will be merged with the
                active configuration except for factory-default in which case
                the active configuration is cleared first.
conn            Display connection information
console         Set idle timeout for the serial console of the PIX
cpu             Display cpu usage
Crashinfo       Read, write and configure crash write to flash.
crypto          Configure IPsec, IKE, and CA
ctiqbe          Show the current data stored for each CTIQBE session.
curpriv         Display current privilege level
debug           Debug packets or ICMP tracings through the PIX Firewall.
dhcpd           Configure DHCP Server
dhcprelay       Configure DHCP Relay Agent
domain-name     Change domain name
```

```
dynamic-map     Specify a dynamic crypto map template
eeprom          show or reprogram the 525 onboard i82559 devices
enable          Configure enable passwords
established     Allow inbound connections based on established connections
failover        Enable/disable PIX failover feature to a standby PIX
filter          Enable, disable, or view URL, FTP, HTTPS, Java, and ActiveX filg
fips-mode       Enable or disable FIPS mode
fixup           Add or delete PIX service and feature defaults
flashfs         Show, destroy, or preserve filesystem information
fragment        Configure the IP fragment database
global          Specify, delete or view global address pools,
                or designate a PAT(Port Address Translated) address
h225            Show the current h225 data stored for each connection.
h245            List the h245 connections.
h323-ras        Show the current h323 ras data stored for each connection.
history         Display the session command history
http            Configure HTTP server
icmp            Configure access for ICMP traffic that terminates at an interface
interface       Set network interface parameters and configure VLANs
igmp            Clear or display IGMP groups
ip              Set the ip address and mask for an interface
                Define a local address pool
                Configure Unicast RPF on an interface
                Configure the Intrusion Detection System
ipsec           Configure IPSEC policy
isakmp          Configure ISAKMP policy
isakmp log     Clear events in the isakmp log buffer
local-host      Display or clear the local host network information
logging         Enable logging facility
mac-list        Add a list of mac addresses using first match search
map             Configure IPsec crypto map
memory          System memory utilization
mgcp            Configure the Media Gateway Control Protocol fixup
mroute          Configure a multicast route
mtu             Specify MTU(Maximum Transmission Unit) for an interface
multicast       Configure multicast on an interface
name            Associate a name with an IP address
nameif          Assign a name to an interface
names           Enable, disable or display IP address to name conversion
nat             Associate a network with a pool of global IP addresses
ntp             Configure Network Time Protocol
object-group    Create an object group for use in 'access-list', 'conduit', etc
ospf            Show OSPF information or clear ospf items.
outbound        Create an outbound access list
pager           Control page length for pagination
passwd          Change Telnet console access password
pdm             Configure Pix Device Manager
prefix-list     Configure a prefix-list
privilege       Configure/Display privilege levels for commands
processes       Display processes
rip             Broadcast default route or passive RIP
route           Enter a static route for an interface
route-map       Create a route-map.
router          Create/configure OSPF routing process
routing         Configure interface specific unicast routing parameters.
running-config  Display the current running configuration
service         Enable system services
session         Access an internal AccessPro router console
shun            Manages the filtering of packets from undesired hosts
sip             Show the current data stored for each SIP session.
skinny          Show the current data stored for each Skinny session.
snmp-server     Provide SNMP and event information
ssh             Add SSH access to PIX console, set idle timeout, display
                list of active SSH sessions & terminate a SSH session
```

```
startup-config  Display the startup configuration
static          Configure one-to-one address translation rule
sysopt          Set system functional option
tcpstat         Display status of tcp stack and tcp connections
tech-support    Tech support
telnet          Add telnet access to PIX console and set idle timeout
terminal        Set terminal line parameters
tftp-server     Specify default TFTP server address and directory
timeout         Set the maximum idle times
traffic         Counters for traffic statistics
uauth           Display or clear current user authorization information
url-cache       Enable URL caching
url-block       Enable URL pending block buffer and long URL support
url-server      Specify a URL filter server
username        Configure user authentication local database
version         Display PIX system software version
virtual         Set address for authentication virtual servers
vpdn            Configure VPDN (PPTP, L2TP, PPPoE) Policy
vpnclient       Configure Easy VPN Remote
vpngroup        Configure group settings for Cisco VPN Clients and
                Cisco Easy VPN Remote products
who             Show active administration sessions on PIX
xlate           Display current translation and connection slot information
```

show blocks/clear blocks

Show system buffer utilization.

show blocks

clear blocks

Syntax Description

blocks	The blocks in the preallocated system buffer.

Command Modes

Privileged mode.

Usage Guidelines

The **show blocks** command lists preallocated system buffer utilization. In the **show blocks** command listing, the SIZE column displays the block type. The MAX column is the maximum number of allocated blocks. The LOW column is the fewest blocks available since last reboot. The CNT column is the current number of available blocks. A zero in the LOW column indicates a previous event where memory exhausted. A zero in the CNT column means memory is exhausted now. Exhausted memory is not a problem as long as traffic is moving through the PIX Firewall. You can use the **show conn** command to see if traffic is moving. If traffic is not moving and the memory is exhausted, a problem may be indicated.

The **clear blocks** command keeps the maximum count to whatever number is allocated in the system and equates the low count to the current count.

You can also view the information from the **show blocks** command using SNMP.

Examples

The following is sample output from the **show blocks** command:

```
show blocks
  SIZE    MAX    LOW    CNT
     4   1600   1600   1600
    80    100     97     97
   256     80     79     79
  1550    788    402    404
 65536      8      8      8
```

show checksum

Display the configuration checksum.

show checksum

Syntax Description

checksum	The hexadecimal numbers that act as a digital summary of the contents of the configuration.

Command Modes

Unprivileged mode.

Usage Guidelines

The **show checksum** command displays four groups of hexadecimal numbers that act as a digital summary of the contents of the configuration. This same information stores with the configuration when you store it in Flash memory. By using the **show config** command and viewing the checksum at the end of the configuration listing and using the **show checksum** command, you can compare the numbers to see if the configuration has changed. The PIX Firewall tests the checksum to determine if a configuration has not been corrupted.

If a dot (".") appears before the checksum in the **show config** or **show checksum** command output, this is a normal configuration load or write mode indicator (when loading from or writing to the firewall Flash memory). This "." is provided to show that the firewall is preoccupied with the operation but not "hung up". It is analogous to a "system processing, please wait" message.

Examples

The following is sample output from the **show checksum** command:

```
show checksum
Cryptochecksum: 1a2833c0 129ac70b 1a88df85 650dbb81
```

show chunkstat

Displays information about management of memory chunks.

show chunkstat

Syntax Description

chunkstat	Displays internal information about management of memory chunks.

Command Modes

Unprivileged mode.

Usage Guidelines

The command **show chunkstat** displays summary information about chunk management, followed by a dump showing the address, content, links, flags and other details.

Examples

The following is sample output from the **show chunkstat** command:

```
show chunkstat
Result of firewall command: "show chunkstat"

Chunk statistics: created 1, destroyed: 0,sibs created: 0, sibs trimmed: 0
Dump of chunk at 0100e09c, name "OSPF redist route node chunks", data start @ 0100e504,
end @ 01010904
 flink: 01008f6c, blink: 01008f6c
 next: cccccccc, next_sibling: 00000000, prev_sibling: 00000000
 flags 00000005
 maximum chunk elt's: 256, elt size: 36, index first free 256
 # chunks in use: 0, HWM of total used: 1, alignment: 8
Chunk statistics: created 1, destroyed: 0,sibs created: 0, sibs trimmed: 0
Dump of chunk at 00eb4cbc, name "ulimit chunk", data start @ 00eb4d8c, end @ 00eb4f8c
 flink: 00578910, blink: 00578910
 next: cccccccc, next_sibling: 00000000, prev_sibling: 00000000
 flags 00000001
 maximum chunk elt's: 32, elt size: 16, index first free 32
 # chunks in use: 0, HWM of total used: 0, alignment: 0
Chunk statistics: created 1, destroyed: 0,sibs created: 0, sibs trimmed: 0
Dump of chunk at 00ea0cd4, name "uauth chunk", data start @ 00ea0da4, end @ 00eb4ca4
 flink: 005793a0, blink: 005793a0
 next: cccccccc, next_sibling: 00000000, prev_sibling: 00000000
 flags 00000001
 maximum chunk elt's: 32, elt size: 2552, index first free 32
 # chunks in use: 0, HWM of total used: 1, alignment: 0
Chunk statistics: created 1, destroyed: 0,sibs created: 0, sibs trimmed: 0
Dump of chunk at 00df706c, name "IP subnet NDB entry", data start @ 00df788c, end @
00e8522c
 flink: 00527914, blink: 00527914
 next: cccccccc, next_sibling: 00000000, prev_sibling: 00000000
 flags 00000009
 maximum chunk elt's: 500, elt size: 1156, index first free 498
 # chunks in use: 2, HWM of total used: 2, alignment: 0
Chunk statistics: created 1, destroyed: 0,sibs created: 0, sibs trimmed: 0
Dump of chunk at 00de56c4, name "IP single NDB entry", data start @ 00de5ee4, end @
00df7054
 flink: 005278f4, blink: 005278f4
 next: cccccccc, next_sibling: 00000000, prev_sibling: 00000000
 flags 00000009
 maximum chunk elt's: 500, elt size: 136, index first free 497
 # chunks in use: 3, HWM of total used: 3, alignment: 0
Chunk statistics: created 1, destroyed: 0,sibs created: 0, sibs trimmed: 0
Dump of chunk at 00dd9f34, name "mroute chunk", data start @ 00dd9fa4, end @ 00dda064
 flink: 0056bf10, blink: 0056bf10
 next: cccccccc, next_sibling: 00000000, prev_sibling: 00000000
 flags 00000001
 maximum chunk elt's: 8, elt size: 24, index first free 8
 # chunks in use: 0, HWM of total used: 0, alignment: 0
Chunk statistics: created 1, destroyed: 0,sibs created: 0, sibs trimmed: 0
Dump of chunk at 00dd76cc, name "radix trie", data start @ 00dd7b1c, end @ 00dd9f1c
 flink: 00dd7684, blink: 00dd7684
 next: cccccccc, next_sibling: 00000000, prev_sibling: 00000000
```

show conn

Display all active connections.

show conn [**count**] | [**detail**] | [**protocol tcp** | **udp** | *protocol*] [{**foreign** | **local**} *ip* [*-ip2*]] [**netmask** *mask*]] [{**lport** | **fport**} *port1* [*-port2*]]

show conn state [**up**] [**,conn_inbound**][**,ctiqbe**][**,data_in**][**,data_out**][**,dump**][**,finin**] [**,finout**][**,h225**][**,h323**][**,http_get**][**,mgcp**][**,nojava**][**,rpc**][**,sip**][**,skinny**][**,smtp_data**] [**,smtp_banner**] [**,sqlnet_fixup_data**][**,smtp_incomplete**]

Syntax Description

count	Display only the number of used connections. The precision of the displayed count may vary depending on traffic volume and the type of traffic passing through the PIX Firewall unit.
detail	If specified, displays translation type and interface information.
{**foreign** | **local**} *ip* [**-***ip2*] **netmask** *mask*	Display active connections by the foreign IP address or by local IP address. Qualify foreign or local active connections by network mask.
fixup	Display whether or not RTP traffic is flowing through the PIX Firewall.
{**lport** | **fport**} *port1* [**-***port2*]	Display foreign or local active connections by port. See "Ports" in Chapter 2, "Using PIX Firewall Commands" for a list of valid port literal names.
protocol tcp | **udp** | *protocol*	Display active connections by protocol type. *protocol* is a protocol specified by number. See "Protocols" in Chapter 2, "Using PIX Firewall Commands" for a list of valid protocol literal names.
state	Display active connections by their current state: up (**up**), inbound connection (**conn_inbound**), Computer Telephony Interface Quick Buffer Encoding (CTIQBE) connection (**ctiqbe**), inbound data (**data_in**), outbound data (**data_out**), dump clean up connection (**dump**), FIN inbound (**finin**), FIN outbound (**finout**), H.225 connection (**h225**), H.323 connection (**h323**), HTTP get (**http_get**), Media Gateway Control Protocol (MGCP) connection (**mgcp**), an **outbound** command denying access to Java applets (**nojava**), RPC connection (**rpc**), SIP connection (**sip**), Skinny Client Control Protocol (SCCP) connection (**skinny**), SMTP mail banner (**smtp_banner**), SMTP mail data (**smtp_data**), SQL*Net data fix up (**sqlnet_fixup_data**), and incomplete SMTP mail connection (**smtp_incomplete**).

Command Modes

Privileged mode.

Usage Guidelines

The **show conn** command displays the number of, and information about, active TCP connections. When specifying multiple **show conn state** options, use commas without spaces to the list states to be displayed. For example, the following is correct:

```
pixfirewall(config)# show conn state up,rpc,h323,sip
```

If you insert spaces, the firewall will not recognize the command.

You can also view the connection count information from the **show conn** command using SNMP.

Note For connections using a DNS server, the source port of the connection may be replaced by the *IP address of DNS server* in the **show conn** output.

The **show conn detail** command displays the following information:

{**UDP** | **TCP**} *outside_ifc*:*real_addr*/*real-port* [(*map_addr*/*port*)] *inside_ifc*:*real_addr*/*real_port* [(*map-addr*/*port*)] **flags** *flags*

The connection flags are defined in Table 8-2.

Table 8-2 Connection Flags

Flag	Description
a	awaiting outside ACK to SYN
A	awaiting inside ACK to SYN
B	initial SYN from outside
C	Computer Telephony Interface Quick Buffer Encoding (CTIQBE) media connection
d	dump
D	DNS
E	outside back connection
f	inside FIN
F	outside FIN
g	Media Gateway Control Protocol (MGCP) connection
G	connection is part of a group[1]
h	H.225
H	H.323
i	incomplete TCP or UDP connection
I	inbound data
k	Skinny Client Control Protocol (SCCP) media connection
m	SIP media connection
M	SMTP data
O	outbound data
p	replicated (unused)
P	inside back connection
q	SQL*Net data
r	inside acknowledged FIN
R	outside acknowledged FIN for TCP connection
R	UDP RPC[2]
s	awaiting outside SYN
S	awaiting inside SYN

Table 8-2 Connection Flags (continued)

Flag	Description
t	SIP transient connection[3]
T	SIP connection[4]
U	up

1. The G flag indicates the connection is part of a group. It is set by the GRE and FTP Strict fixups to designate the control connection and all its associated secondary connections. If the control connection terminates, then all associated secondary connections are also terminated.
2. Because each row of **show conn** command output represents one connection (TCP or UDP), there will be only one R flag per row.
3. For UDP connections, the value t indicates that it will timeout after one minute.
4. For UDP connections, the value T indicates that the connection will timeout according to the value specified using the **timeout sip** command.

Examples

The following example shows a TCP session connection from inside host 10.1.1.15 to the outside Telnet server at 192.150.49.10. Because there is no B flag, the connection is initiated from the inside. The “U”, “I”, and “O” flags denote that the connection is active and has received inbound and outbound data.

```
pixfirewall(config)# show conn
2 in use, 2 most used
TCP out 192.150.49.10:23 in 10.1.1.15:1026 idle 0:00:22
Bytes 1774 flags UIO
UDP out 192.150.49.10:31649 in 10.1.1.15:1028 idle 0:00:14
flags D-
```

The following example shows a UDP connection from outside host 192.150.49.10 to inside host 10.1.1.15. The D flag denotes that this is a DNS connection. The number 1028 is the DNS ID over the connection.

```
pixfirewall(config)# show conn detail
2 in use, 2 most used
Flags: A - awaiting inside ACK to SYN, a - awaiting outside ACK to SYN,
       B - initial SYN from outside, C - CTIBQE media, D - DNS, d - dump,
       E - outside back connection, f - inside FIN, F - outside FIN,
       G - group, g - MGCP, H - H.323, h - H.255.0, I - inbound data, i - incomplete,
        k - Skinny media, M - SMTP data, m - SIP media
       O - outbound data, P - inside back connection,
       q - SQL*Net data, R - outside acknowledged FIN,
       R - UDP RPC, r - inside acknowledged FIN, S - awaiting inside SYN,
        s - awaiting outside SYN, T - SIP, t - SIP transient, U - up
TCP outside:192.150.49.10/23 inside:10.1.1.15/1026 flags UIO
UDP outside:192.150.49.10/31649 inside:10.1.1.15/1028 flags dD
```

The following is sample output from the **show conn** command:

```
show conn
6 in use, 6 most used
  TCP out 209.165.201.1:80 in 10.3.3.4:1404 idle 0:00:00 Bytes 11391
  TCP out 209.165.201.1:80 in 10.3.3.4:1405 idle 0:00:00 Bytes 3709
  TCP out 209.165.201.1:80 in 10.3.3.4:1406 idle 0:00:01 Bytes 2685
  TCP out 209.165.201.1:80 in 10.3.3.4:1407 idle 0:00:01 Bytes 2683
  TCP out 209.165.201.1:80 in 10.3.3.4:1403 idle 0:00:00 Bytes 15199
  TCP out 209.165.201.1:80 in 10.3.3.4:1408 idle 0:00:00 Bytes 2688
  UDP out 209.165.201.7:24 in 10.3.3.4:1402 idle 0:01:30
  UDP out 209.165.201.7:23 in 10.3.3.4:1397 idle 0:01:30
  UDP out 209.165.201.7:22 in 10.3.3.4:1395 idle 0:01:30
```

In this example, host 10.3.3.4 on the inside has accessed a website at 209.165.201.1. The global address on the outside interface is 209.165.201.7.

show cpu usage

The **show cpu usage** command displays CPU utilization.

show cpu usage

Syntax Description		
	cpu usage	The central processing unit (CPU) usage data.

Command Modes Privileged or configuration mode.

Usage Guidelines The **show cpu usage** command displays the central processing unit (CPU) usage information.

Examples The following is sample output from the **show cpu usage** command:

```
pixfirewall# show cpu usage
CPU utilization for 5 seconds: p1%; 1 minute: p2%; 5 minutes: p3%
```

The percentage usage prints as NA (not applicable) if the usage is unavailable for the specified time interval. This can happen if the user asks for CPU usage before the 5-second, 1-minute, or 5-minute time interval has elapsed.

show crypto engine [verify]

Shows cryptography engine statistics or runs the Known Answer Test (KAT).

show crypto engine [**verify**]

Syntax Description		
	crypto engine	Displays usage statistics for the firewall cryptography engine.
	verify	Runs the Known Answer Test (KAT).

Command Modes Privileged or configuration mode.

Usage Guidelines The **show crypto engine** command displays usage statistics for the cryptography engine used by the firewall.

The **show crypto engine verify** command runs the Known Answer Test (KAT) from the firewall CLI. Additionally, when booted for the first time or after a reload, the firewall performs the Know Answer Test (KAT) before any configuration information is read from the Flash memory. If the KAT fails, then the firewall issues an error message and reloads. The KAT is performed to check the integrity of the cryptography engine used by the firewall.

Examples

The following example shows sample output for the **show crypto engine** command:

```
pixfirewall# show crypto engine
Crypto Engine Connection Map:
    size = 8, free = 6, used = 1, active = 1
```

In this command output, *size* is total number of undirectional IPSec tunnels, *free* is the number of unused undirectional IPSec tunnels, *used* is the number of allocated undirectional IPSec tunnels, and *active* is the number of active undirectional IPSec tunnels. Because tunnel 0 is reserved for system use, *size* is equal to *free* plus *used* plus one.

The following example shows sample output for the **show crypto engine** command when output is specified for a VAC or a VAC+:

```
VAC+:

pixfirewall# show crypto interface
Encryption hardware device : VAC+ (Crypto5823 revision 0x1)

VAC:

pixfirewall# show crypto interface
Encryption hardware device : VAC (IRE2141 with 2048KB, HW:1.0, CGXROM:1.9, FW:6.5)
```

The following example shows the **show crypto engine verify** command output for a successful KAT:

```
pixfirewall# show crypto engine verify
FIPS: Known Answer Test begin

FIPS: software DES      success
FIPS: software SHA      success
FIPS: software RSA      success

FIPS:    software to software       DES/SHA1 tunnel check success.

FIPS: Known Answer Test finish
```

The following is sample output from a KAT that failed during start up of the firewall:

```
Cisco PIX Firewall Version 6.3(1)
Licensed Features:
Failover: Enabled
VPN-DES: Enabled
VPN-3DES-AES: Enabled
Maximum Interfaces: 6
Cut-through Proxy: Enabled
Guards: Enabled
URL-filtering: Enabled
Inside Hosts: Unlimited
Throughput: Unlimited
IKE peers: Unlimited

This PIX has an Unrestricted (UR) license.

FIPS: software AES fail
```

```
An internal error occurred. Specifically, a programming assertion was
violated. Copy the error message exactly as it appears, and get the
output of the show version command and the contents of the configuration
file. Then call your technical support representative.

assertion "result != FALSE" failed: file "crypto_nist_tests.c", line 529
No thread name

Traceback:
0: 0040d84d
1: 00260608
...
```

show crypto interface [counters]

Displays the VPN accelerator cards (VACs) installed in the firewall chassis and, for the VAC+, the packet, payload byte, queue length, and moving average counters for traffic moving through the card.

show crypto interface [**counters**]

clear crypto interface counters

Syntax Description

counters	Displays packet count, byte queue, and moving averages for traffic through a VAC+.
crypto interface	Displays the VPN accelerator cards (VACs) installed in the firewall chassis.

Command Modes

Privileged or configuration mode.

Usage Guidelines

The **show crypto interface** command lists VPN accelerator cards (VACs) installed in the firewall chassis. (This same information is also displayed in **show version** output.)

The **show crypto interface counters** command displays information, as described in Table 8-3, for the PIX Firewall VAC+ card only.

Table 8-3 show crypto interface counters

Counter	Description
`interfaces`	The number and type of crypto interface cards installed.
`packet count`	The number of packets sent to the installed crypto interface card(s).
`payload bytes`	The number of bytes of payload either after decapsulation or before encapsulation.
`input queue (curr/max)`	The total number of packets that are awaiting service from the crypto interface card(s).
`interface queue (curr/max)`	The total number of packets that have been queued at the crypto interface card(s) for service.

Table 8-3 show crypto interface counters

Counter	Description
output queue (curr/max)	The total number of packets that have been released by the crypto interface card(s) and are awaiting dispatch to the packet path.
moving averages 5second 1minute 5minute	5 second, 1 minute, and 5 minute moving averages of the packet count and payload bytes through all crypto interface cards.

The **clear crypto interface counters** command clears only the packet, payload byte, queue length, and moving average counters. It does not affect any actual packets queued.

Examples

The following is sample output from the **show crypto interface** and **show crypto interface counters** commands when a VAC+ card is installed:

```
pixfirewall# show crypto interface
Encryption hardware device : Crypto5823 (revision 0x1)
pixfirewall(config)# show crypto interface counters

interfaces: 1
  Crypto5823 (revision 0x1), maximum queue size 64

packet count:               318657093
payload bytes:              89861300946
input     queue (curr/max): 1336/1584
interface queue (curr/max): 64/64
output    queue (curr/max): 0/64

moving averages
  5second   128273 pkts/sec   289 Mbits/sec
  1minute   128326 pkts/sec   290 Mbits/sec
  5minute   128279 pkts/sec   289 Mbits/sec
```

The following is the same sample output after the **clear crypto interface counters** command has been used:

```
pixfirewall# clear crypto interface counters
pixfirewall# show crypto interface counters
interfaces: 1
  Crypto5823 (revision 0x1), maximum queue size 64

packet count:               355968
payload bytes:              100382976
input     queue (curr/max): 1317/1537
interface queue (curr/max): 64/64
output    queue (curr/max): 0/64

moving averages
  5second       NA pkts/sec    NA Mbits/sec
  1minute       NA pkts/sec    NA Mbits/sec
  5minute       NA pkts/sec    NA Mbits/sec
```

The following is sample output from the **show crypto interface** and **show crypto interface counters** commands when a VAC card is installed:

```
pixfirewall# show crypto interface
Encryption hardware device : IRE2141 with 2048KB, HW:1.0, CGXROM:1.9, FW:6.5
pixfirewall# show crypto interface counters
```

```
no crypto interface counters available
```

The following is sample output from the **show crypto interface** and **show crypto interface counters** commands when no crypto interface card is installed (neither a VAC nor a VAC+):

```
pixfirewall# show crypto interface
pixfirewall# show crypto interface counters
no crypto interface counters available
```

show ip local pool

The **show ip local pool** command displays:

- any included netmask if it is configured.
- fixes an alignment problem if present with possible varied length pool names.

Syntax Description

ip local pool	List of configured local pool IP addresses

Command Modes

Configuration mode.

Usage Guidelines

The **show ip local pool** command can now output the netmask if it is configured. The **show ip local pool** command displays previously configured local pool addresses.

The following is sample output from the **show ip local pool** command:

```
argus-520(config)# sh ip local pool
Pool            Begin           End             Mask            Free    In use
VPNClient       10.1.0.1        10.1.0.25       Not configured    25         0
...
Pool            Begin           End             Mask            Free    In use
ReallyReallyReallyLongPoolName
                192.168.0.1     192.168.64.0    255.255.0.0     16384         0
```

show history

Display previously entered commands.

show history

Syntax Description

history	The list of previous entries.

Command Modes

Available in unprivileged mode, privileged mode, and configuration mode.

Usage Guidelines The **show history** command displays previously entered commands. You can examine commands individually with the up and down arrows or by entering **^p** to view previously entered lines or **^n** to view the next line.

Examples The following is sample output from the **show history** command when run in unprivileged mode:

```
pixfirewall> show history
    show history
    help
    show history
```

The following is sample output from the **show history** command when run in privileged mode:

```
pixfirewall# show history
    show history
    help
    show history
    enable
    show history
```

The following is sample output from the **show history** command when run in configuration mode:

```
pixfirewall(config)# show history
    show history
    help
    show history
    enable
    show history
    config t
    show history
```

show local-host/clear local host

View local host network states.

show local-host [*ip_address*]

clear local-host [*ip_address*]

Syntax Description

ip_address	Local host IP address.

Command Modes Privileged mode for the **show** commands and configuration mode for the **clear** commands.

Usage Guidelines The **show local-host** command displays the translation and connection slots for all local hosts. This command also provides information for hosts configured with the **nat 0** command when normal translation and connection states may not apply. The **show local-host detail** command displays more information about active xlates and connections. Use the *ip_address* option to limit the display to a single host.

The **clear local-host** command stops traffic on all local hosts. The **clear local-host** *ip_address* command stops traffic on the local host specified by its IP address.

On a PIX 501, cleared hosts are released from the license limit. You can view the number of hosts that are counted toward the license limit with the **show local-host** command.

Note Clearing the network state of a local host stops all connections and xlates associated with the local hosts.

Examples

The following is sample output from the **show local-host** command:

```
show local-host 10.1.1.15
local host: <10.1.1.15>, conn(s)/limit = 2/0, embryonic(s)/limit = 0/0
   Xlate(s):
       PAT Global 172.16.3.200(1024) Local 10.1.1.15(55812)
       PAT Global 172.16.3.200(1025) Local 10.1.1.15(56836)
       PAT Global 172.16.3.200(1026) Local 10.1.1.15(57092)
       PAT Global 172.16.3.200(1027) Local 10.1.1.15(56324)
       PAT Global 172.16.3.200(1028) Local 10.1.1.15(7104)
   Conn(s):
       TCP out 192.150.49.10:23 in 10.1.1.15:1246 idle 0:00:20 Bytes 449 flags UIO
       TCP out 192.150.49.10:21 in 10.1.1.15:1247 idle 0:00:10 Bytes 359 flags UIO
```

The xlate describes the translation slot information and the Conn is the connection state information.

The following is sample command output from the **show local-host** command:

```
pixfirewall(config)# show local-host
local host: <10.1.1.15>, conn(s)/limit = 2/0, embryonic(s)/limit = 0/0
    Xlate(s):
        PAT Global 192.150.49.1(1024) Local 10.1.1.15(516)
        PAT Global 192.150.49.1(0) Local 10.1.1.15 ICMP id 340
        PAT Global 192.150.49.1(1024) Local 10.1.1.15(1028)
    Conn(s):
        TCP out 192.150.49.10:23 in 10.1.1.15:1026 idle 0:00:25
                  Bytes 1774 flags UIO
        UDP out 192.150.49.10:31649 in 10.1.1.15:1028 idle 0:00:17
                  flags D-
```

For comparison, the following is sample command output from the **show local-host detail** command:

```
pixfirewall(config)# show local-host detail
     local host: <10.1.1.15>,
         TCP connection count/limit = 0/unlimited
         TCP embryonic count = 0
         TCP intercept watermark = unlimited
         UDP connection count/limit = 0/unlimited
         Xlate(s):
             TCP PAT from inside:10.1.1.15/1026 to outside:192.150.49.1/1024
                       flags ri
             ICMP PAT from inside:10.1.1.15/21505 to outside:192.150.49.1/0
                       flags ri
             UDP PAT from inside:10.1.1.15/1028 to outside:192.150.49.1/1024
                       flags ri
         Conn(s):
             TCP outside:192.150.49.10/23 inside:10.1.1.15/1026 flags UIO
             UDP outside:192.150.49.10/31649 inside:10.1.1.15/1028 flags dD
```

The next example shows how the **clear local-host** command clears the local host information:

```
clear local-host 10.1.1.15
show local-host 10.1.1.15
```

Once the information is cleared, nothing more displays until the hosts reestablish their connections, which were stopped by the **clear local-host** command, and more data is produced.

show memory

Show system memory utilization.

show memory[*detail*]

Syntax Description

memory	The system memory data.
detail	Additional detail on system memory data.

Command Modes

Privileged mode.

Usage Guidelines

The **show memory** command displays a summary of the maximum physical memory and current free memory available to the PIX Firewall operating system. Memory in the PIX Firewall is allocated as needed.

You can also view the information from the **show memory** command using SNMP.

Examples

The following is sample output from the **show memory** command:

```
Result of firewall command: "show memory"

Free memory:        17149656 bytes
Used memory:        16404776 bytes
-------------     ----------------
Total memory:       33554432 bytes
```

The following is sample output from the **show memory detail** command:

```
Result of firewall command: "show memory detail"

Free memory:                      49734280 bytes
Used memory:
     Allocated memory in use:     11175212 bytes
     Reserved memory:              6199372 bytes
----------------------------   ----------------
Total memory:                     67108864 bytes
----- fragmented memory statistics -----
 fragment size       count          total
    (bytes)                       (bytes)
---------------  ----------  --------------
           16             9             144
           24             2              48
           80             2             160
          128             1             128
          216             1             216
          224             3             672
          232             1             232
          240             1             240
```

```
          296             2            592
          312             1            312
          320             1            320
          816             1            816
         1136             1           1136
         1336             1           1336
         5504             3          16528
         5784             4          23280
         9024             1           9024
        13744             1          13744*
        14768             4          61400
        21872             1          21872
     49582080             1       49582080**
*  - top most releasable chunk.
** - contiguous memory on top of heap.
----- allocated memory statistics -----
 fragment size       count          total
    (bytes)                        (bytes)
---------------  ----------  --------------
           24           20           480
           32         1426         45632
           40         1439         57560
           48          197          9456
           56         1013         56728
           64          110          7040
           72           69          4968
           80           35          2800
           88           56          4928
           96           10           960
          104           12          1248
          112           15          1680
          120           19          2280
          128            7           896
          136           35          4760
          144            3           432
          152           61          9272
          160            7          1120
          168            1           168
          176            1           176
          184            1           184
          200            6          1200
          216            1           216
          224            4           896
          256            1           256
          264          706        186384
          272            2           544
          280            4          1120
          288            2           576
          296            1           296
          304            1           304
          328            1           328
          344            2           688
          352           62         21824
          360            3          1080
          392            1           392
          424            4          1696
          464            1           464
          512           21         10752
          576            4          2304
          640            1           640
          704            2          1408
          768            2          1536
          832            1           832
          896            1           896
```

```
  1024            7          7168
  1088            1          1088
  1152            2          2304
  1408            6          8448
  1536            1          1536
  1600            3          4800
  1664            1          1664
  1856            1          1856
  1920            1          1920
  2048            1          2048
  2112            1          2112
  2240            1          2240
  2368            1          2368
  3072           16         49152
  4096           51        208896
  4608            1          4608
  8192           14        114688
  9728            1          9728
 10240            1         10240
 10752            1         10752
 14848            3         44544
 18944           28        530432
 23040            1         23040
 27136            2         54272
 31232            4        124928
 39424            3        118272
 76288           15       1144320
141824            7        992768
174592            4        698368
436736           10       4367360
698880            3       2096640
```

show ospf

Displays general information about OSPF routing processes.

show ospf [*pid*]

Syntax Description

pid	The ID of the OSPF process.

Defaults

The default is to list all OSPF processes if no *pid* is specified.

Command Modes

The **show ospf** command is available in privileged mode.

Usage Guidelines

The OSPF routing-related **show** commands are available in privileged command mode on the firewall. You do not need to be in an OSPF configuration subcommand mode to use the OSPF-related **show** commands.

If the *pid* is included, only information for the specified routing process is included.

Examples

The following examples are sample output from the **show ospf** [*pidpid*] (with a *pid* of 5)and **show ospf** commands:

```
pixfirewall# show ospf 5

 Routing Process "ospf 5" with ID 127.0.0.1 and Domain ID 0.0.0.5
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
 Number of external LSA 0. Checksum Sum 0x     0
 Number of opaque AS LSA 0. Checksum Sum 0x     0
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 0. 0 normal 0 stub 0 nssa
 External flood list length 0

pixfirewall# show ospf

 Routing Process "ospf 5" with ID 127.0.0.1 and Domain ID 0.0.0.5
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
 Number of external LSA 0. Checksum Sum 0x     0
 Number of opaque AS LSA 0. Checksum Sum 0x     0
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 0. 0 normal 0 stub 0 nssa
 External flood list length 0

 Routing Process "ospf 12" with ID 172.23.59.232 and Domain ID 0.0.0.12
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
 Number of external LSA 0. Checksum Sum 0x     0
 Number of opaque AS LSA 0. Checksum Sum 0x     0
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 0. 0 normal 0 stub 0 nssa
 External flood list length 0
```

Related Commands

prefix-list	Configures a prefix list to be used for OSPF routing.
route-map	Creates a route map for redistributing routes from one routing protocol to another. Used in configuring OSPF routing on the firewall.
router ospf	Configures global parameters for the OSPF routing processes on the firewall, and enables or disables OSPF routing through the firewall.
routing interface	Configures interface-specific OSPF routing parameters.

show ospf border-routers

Displays the internal OSPF routing table entries to an Area Border Router (ABR) and Autonomous System Boundary Router (ASBR).

show ospf border-routers

Syntax Description

border-routers	Area Border Routers (ABRs) and Autonomous System Boundary Routers (ASBRs).

Defaults

None.

Command Modes

The **show ospf border-routers** command is available in privileged mode.

Usage Guidelines

The OSPF routing-related **show** commands are available in privileged command mode on the firewall. You do not need to be in an OSPF configuration subcommand mode to use the OSPF-related **show** commands.

Examples

The following is sample output from the **show ospf border-routers** command:

```
pixfirewall# show ospf border-routers
OSPF Process 109 internal Routing Table
Destination Next Hop Cost Type Rte Type Area SPF No
192.168.97.53 172.16.1.53 10 ABR INTRA 0.0.0.3 3
192.168.103.51 192.168.96.51 10 ABR INTRA 0.0.0.3 3
192.168.103.52 192.168.96.51 20 ASBR INTER 0.0.0.3 3
192.168.103.52 172.16.1.53 22 ASBR INTER 0.0.0.3 3
```

Related Commands

prefix-list	Configures a prefix list to be used for OSPF routing.
route-map	Creates a route map for redistributing routes from one routing protocol to another. Used in configuring OSPF routing on the firewall.
router ospf	Configures global parameters for the OSPF routing processes on the firewall, and enables or disables OSPF routing through the firewall.
routing interface	Configures interface-specific OSPF routing parameters.

show ospf database

Displays LSA information in the OSPF database for a specific network area or router.

show ospf [*pid*] **database** [**internal**] [**adv-router** [*ip_address*]]

show ospf [*pid* [*area_id*]] **database** [**internal**] [**self-originate**] [*lsid*]

show ospf [*pid* [*area_id*]] **database** {**router** | **network** | **summary** | **asbr-summary** | **external** | **nssa-external** | **database-summary**}]

Syntax Description

adv-router [*ip_address*]	Displays all the link-state advertisements (LSAs) of the specified router. If no IP address is included, the information is about the local router itself (in that case, the output is the same as with the **self-originate** keyword).
area_id	The ID of the area that is associated with the OSPF address range. If you intend to associate areas with IP subnets, you can specify a subnet address as the *area_id*. When used in the context of authentication, *area_id* is the identifier of the area on which authentication is to be enabled. When using a cost context, *area_id* is the identifier for the stub or NSSA. When used in the context of a prefix list, *area_id* is the identifier of the area on which filtering is configured. When used in a stub area or not-so-stubby area (NSSA) context, *area_id* is the identifier for the stub or NSSA area. When used in the context of an area range, *area_id* is the identifier of the area at whose boundary to summarize routes.
asbr-summary	Displays information only about the Autonomous System Boundary Router (ASBR) summary LSAs.
database-summary	Displays how many of each type of LSA for each area there are in the database, and the total.
external	Routes external to a specified autonomous system.
internal	Routes that are internal to a specified autonomous system.
ip_address	The IP address of the OSPF router.
lsid	The link state ID, specified as an IP address. The *lsid* describes the portion of the Internet environment that is being described by the link-state advertisement (LSA). The value entered depends on the type of the LSA, but the value must be entered in the form of an IP address, as follows: • When the LSA is describing a network, set *lsid* to the network IP address (for Type 3 summary link advertisements and for autonomous system external link advertisements) or a derived IP address with the network subnet mask (from which the OSPF process interprets the network IP address). • When the LSA is describing a router, set *lsid* to the OSPF router ID of the router. • When an autonomous system external advertisement (Type 5) is describing a default route, set *lsid* to the default destination (0.0.0.0).
network	Displays information only about the network LSAs.
nssa-external	Displays information only about the not-so-stubby area (NSSA) external LSAs.
pid	The ID of the OSPF process.
router	Displays information only about the router LSAs.
self-originate	Displays only self-originated LSAs (from the local router).
summary	Displays information only about the summary LSAs.

Defaults None.

Command Modes The **show ospf database** command is available in privileged mode.

Usage Guidelines The OSPF routing-related **show** commands are available in privileged command mode on the firewall. You do not need to be in an OSPF configuration subcommand mode to use the OSPF-related **show** commands.

The various forms of this command deliver information about different OSPF link-state advertisements (LSAs).

Examples The following is sample output from the **show ospf database** command:

```
pixfirewall# show ospf database
OSPF Router with ID(192.168.1.11) (Process ID 1)
                Router Link States(Area 0)
Link ID   ADV Router   Age   Seq# Checksum Link count
192.168.1.8 192.168.1.8 1381 0x8000010D    0xEF60 2
192.168.1.11 192.168.1.11 1460 0x800002FE    0xEB3D 4
192.168.1.12 192.168.1.12 2027 0x80000090    0x875D 3
192.168.1.27 192.168.1.27 1323 0x800001D6    0x12CC 3
                Net Link States(Area 0)
Link ID ADV Router   Age   Seq# Checksum
172.16.1.27 192.168.1.27 1323 0x8000005B    0xA8EE
172.17.1.11 192.168.1.11 1461 0x8000005B    0x7AC
                Type-10 Opaque Link Area Link States (Area 0)
Link ID ADV Router   Age Seq# Checksum Opaque ID
10.0.0.0 192.168.1.11 1461 0x800002C8    0x8483   0
10.0.0.0 192.168.1.12 2027 0x80000080    0xF858   0
10.0.0.0 192.168.1.27 1323 0x800001BC    0x919B   0
10.0.0.1 192.168.1.11 1461 0x8000005E    0x5B43   1
```

The following is sample output from the **show ospf database asbr-summary** command:

```
pixfirewall# show ospf database asbr-summary
OSPF Router with ID(192.168.239.66) (Process ID 300)
Summary ASB Link States(Area 0.0.0.0)
Routing Bit Set on this LSA
LS age: 1463
Options: (No TOS-capability)
LS Type: Summary Links(AS Boundary Router)
Link State ID: 172.16.245.1 (AS Boundary Router address)
Advertising Router: 172.16.241.5
LS Seq Number: 80000072
Checksum: 0x3548
Length: 28
Network Mask: 0.0.0.0
TOS: 0 Metric: 1
```

The following is sample output from the **show ospf database router** command:

```
pixfirewall# show ospf database router
OSPF Router with id(192.168.239.66) (Process ID 300)
Router Link States(Area 0.0.0.0)
Routing Bit Set on this LSA
LS age: 1176
Options: (No TOS-capability)
```

```
LS Type: Router Links
Link State ID: 10.187.21.6
Advertising Router: 10.187.21.6
LS Seq Number: 80002CF6
Checksum: 0x73B7
Length: 120
AS Boundary Router
Number of Links: 8
Link connected to: another Router (point-to-point)
(link ID) Neighboring Router ID: 10.187.21.5
(Link Data) Router Interface address: 10.187.21.6
Number of TOS metrics: 0
TOS 0 Metrics: 2
```

The following is sample output from the **show ospf database network** command:

```
pixfirewall# show ospf database network
OSPF Router with id(192.168.239.66) (Process ID 300)
Displaying Net Link States(Area 0.0.0.0)
LS age: 1367
Options: (No TOS-capability)
LS Type: Network Links
Link State ID: 10.187.1.3 (address of Designated Router)
Advertising Router: 192.168.239.66
LS Seq Number: 800000E7
Checksum: 0x1229
Length: 52
Network Mask: 255.255.255.0
Attached Router: 192.168.239.66
Attached Router: 10.187.241.5
Attached Router: 10.187.1.1
Attached Router: 10.187.54.5
Attached Router: 10.187.1.5
```

The following is sample output from the **show ospf database summary** command:

```
pixfirewall# show ospf database summary
OSPF Router with id(192.168.239.66) (Process ID 300)
Displaying Summary Net Link States(Area 0.0.0.0)
LS age: 1401
Options: (No TOS-capability)
LS Type: Summary Links(Network)
Link State ID: 10.187.240.0 (summary Network Number)
Advertising Router: 10.187.241.5
LS Seq Number: 80000072
Checksum: 0x84FF
Length: 28
Network Mask: 255.255.255.0 TOS: 0 Metric: 1
```

The following is sample output from the **show ospf database external** command:

```
pixfirewall# show ospf database external
OSPF Router with id(192.168.239.66) (Autonomous system 300)
                  Displaying AS External Link States
LS age: 280
Options: (No TOS-capability)
LS Type: AS External Link
Link State ID: 143.10.0.0 (External Network Number)
Advertising Router: 10.187.70.6
LS Seq Number: 80000AFD
Checksum: 0xC3A
Length: 36
Network Mask: 255.255.0.0
      Metric Type: 2 (Larger than any link state path)
      TOS: 0
```

```
Metric: 1
Forward Address: 0.0.0.0
External Route Tag: 0
```

Related Commands

prefix-list	Configures a prefix list to be used for OSPF routing.
route-map	Creates a route map for redistributing routes from one routing protocol to another. Used in configuring OSPF routing on the firewall.
router ospf	Configures global parameters for the OSPF routing processes on the firewall, and enables or disables OSPF routing through the firewall.
routing interface	Configures interface-specific OSPF routing parameters.

show ospf flood-list

Displays a list of OSPF link-state advertisements (LSAs) waiting to be flooded over an interface.

show ospf flood-list *if_name*

Syntax Description

flood-list	The list of link-state advertisements (LSAs) waiting to be flooded over an interface.
if_name	The name of the interface for which to display neighbor information.

Defaults

None.

Command Modes

The **show ospf flood-list** command is available in privileged mode.

Usage Guidelines

The OSPF routing-related **show** commands are available in privileged command mode on the firewall. You do not need to be in an OSPF configuration subcommand mode to use the OSPF-related **show** commands.

Examples

The following is sample output from the **show ospf flood-list** command, where the *if_name* is **outside**:

```
pixfirewall# show ospf flood-list outside
Interface outside, Queue length 20
Link state flooding due in 12 msec
Type LS ID  ADV RTR   Seq NO Age Checksum
5 10.2.195.0 192.168.0.163 0x80000009 0 0xFB61
5 10.1.192.0 192.168.0.163 0x80000009 0 0x2938
5 10.2.194.0 192.168.0.163 0x80000009 0 0x757
5 10.1.193.0 192.168.0.163 0x80000009 0 0x1E42
5 10.2.193.0 192.168.0.163 0x80000009 0 0x124D
5 10.1.194.0 192.168.0.163 0x80000009 0 0x134C
```

Related Commands

prefix-list	Configures a prefix list to be used for OSPF routing.
route-map	Creates a route map for redistributing routes from one routing protocol to another. Used in configuring OSPF routing on the firewall.
router ospf	Configures global parameters for the OSPF routing processes on the firewall, and enables or disables OSPF routing through the firewall.
routing interface	Configures interface-specific OSPF routing parameters.

show ospf interface

Displays OSPF-related interface information.

show ospf interface *if_name*

Syntax Description

if_name	The name of the interface for which to display OSPF-related information.

Defaults

None.

Command Modes

The **show ospf interface** *if_name* command is available in privileged mode.

Usage Guidelines

The OSPF routing-related **show** commands are available in privileged command mode on the firewall. You do not need to be in an OSPF configuration subcommand mode to use the OSPF-related **show** commands.

Examples

The following is sample output from the **show ospf interface** *if_name* command, where the *if_name* is **inside**:

```
pixfirewall# show ospf interface inside
inside is up, line protocol is up
Internet Address 192.168.254.202, Mask 255.255.255.0, Area 0.0.0.0
AS 201, Router ID 192.77.99.1, Network Type BROADCAST, Cost: 10
Transmit Delay is 1 sec, State OTHER, Priority 1
Designated Router id 192.168.254.10, Interface address 192.168.254.10
Backup Designated router id 192.168.254.28, Interface addr 192.168.254.28
Timer intervals configured, Hello 10, Dead 60, Wait 40, Retransmit 5
Hello due in 0:00:05
Neighbor Count is 8, Adjacent neighbor count is 2
  Adjacent with neighbor 192.168.254.28 (Backup Designated Router)
  Adjacent with neighbor 192.168.254.10 (Designated Router)
```

Related Commands

prefix-list	Configures a prefix list to be used for OSPF routing.
route-map	Creates a route map for redistributing routes from one routing protocol to another. Used in configuring OSPF routing on the firewall.

router ospf	Configures global parameters for the OSPF routing processes on the firewall, and enables or disables OSPF routing through the firewall.
routing interface	Configures interface-specific OSPF routing parameters.

show ospf neighbor

Displays OSPF-neighbor information on a per-interface basis.

show ospf neighbor [*if_name*] [*nbr_router_id*] [**detail**]

Syntax Description

detail	List all neighbors.
if_name	The name of the interface for which to display neighbor information.
nbr_router_id	The IP address of the neighbor router.

Defaults

None.

Command Modes

The **show ospf neighbor** command is available in privileged mode.

Usage Guidelines

The OSPF routing-related **show** commands are available in privileged command mode on the firewall. You do not need to be in an OSPF configuration subcommand mode to use the OSPF-related **show** commands.

Examples

The following is sample output from the **show ospf neighbor** *if_name nbr_router_id* command, where the *if_name* is **inside** and the *nbr_router_id* is 10.199.199.137:

```
pixfirewall# show ospf neighbor inside 10.199.199.137
Neighbor 10.199.199.137, interface address 192.168.80.37
In the area 0.0.0.0 via interface inside
Neighbor priority is 1, State is FULL
Options 2
Dead timer due in 0:00:37
Link State retransmission due in 0:00:04
```

The following is sample output from the **show ospf neighbor detail** command, where the *if_name* is **outside**:

```
pixfirewall# show ospf neighbor outside detail
Neighbor 192.168.5.2, interface address 10.225.200.28
In the area 0 via interface outside
Neighbor priority is 1, State is FULL, 6 state changes
DR is 10.225.200.28 BDR is 10.225.200.30
Options is 0x42
Dead timer due in 00:00:36
Neighbor is up for 00:09:46
Index 1/1, retransmission queue length 0, number of retransmission 1
First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
Last retransmission scan length is 1, maximum is 1
```

```
Last retransmission scan time is 0 msec, maximum is 0 msec
```

Related Commands

prefix-list	Configures a prefix list to be used for OSPF routing.
route-map	Creates a route map for redistributing routes from one routing protocol to another. Used in configuring OSPF routing on the firewall.
router ospf	Configures global parameters for the OSPF routing processes on the firewall, and enables or disables OSPF routing through the firewall.
routing interface	Configures interface-specific OSPF routing parameters.

show ospf request-list

Displays a list of all link-state advertisements (LSAs) requested by a router.

show ospf request-list *nbr_router_id if_name*

Syntax Description

if_name	The name of the interface for which to display neighbor information. Displays the list of all LSAs requested by the router from this interface.
nbr_router_id	The ID of the neighbor router, specified by IP address. Displays the list of all LSAs requested by the router from this neighbor.

Defaults

None.

Command Modes

The **show ospf request-list** *nbr_router_id if_name* command is available in privileged mode.

Usage Guidelines

The OSPF routing-related **show** commands are available in privileged command mode on the firewall. You do not need to be in an OSPF configuration subcommand mode to use the OSPF-related **show** commands.

Examples

The following is sample output from the **show ospf request-list** command, where the *nbr_router_id* is 192.168.1.12 and the *if_name* is **inside**:

```
pixfirewall# show ospf request-list 192.168.1.12 inside
OSPF Router with ID (192.168.1.11) (Process ID 1)
Neighbor 192.168.1.12, interface inside address 172.16.1.12
Type LS ID   ADV RTR   Seq NO Age Checksum
1 192.168.1.12 192.168.1.12 0x8000020D 8 0x6572
```

Related Commands

prefix-list	Configures a prefix list to be used for OSPF routing.
route-map	Creates a route map for redistributing routes from one routing protocol to another. Used in configuring OSPF routing on the firewall.

router ospf	Configures global parameters for the OSPF routing processes on the firewall, and enables or disables OSPF routing through the firewall.
routing interface	Configures interface-specific OSPF routing parameters.

show ospf retransmission-list

Displays a list of all link-state advertisements (LSAs) waiting to be resent.

show retransmission-list *nbr_router_id if_name*

Syntax Description

if_name	The name of the interface for which to display neighbor information. Displays the list of all LSAs waiting to be resent for this neighbor.
nbr_router_id	The ID of the neighbor router, specified by IP address. Displays the list of all LSAs waiting to be resent for this interface.

Defaults

None.

Command Modes

The **show retransmission-list** *nbr_router_id if_name* command is available in privileged mode.

Usage Guidelines

The OSPF routing-related **show** commands are available in privileged command mode on the firewall. You do not need to be in an OSPF configuration subcommand mode to use the OSPF-related **show** commands.

Examples

The following is sample output from the **show ospf retransmission-list** command, where the *nbr_router_id* is 192.168.1.11 and the *if_name* is **outside**:

```
pixfirewall# show ospf retransmission-list 192.168.1.11 outside
OSPF Router with ID (192.168.1.12) (Process ID 1)
Neighbor 192.168.1.11, interface outside address 172.16.1.11
Link state retransmission due in 3764 msec, Queue length 2
Type LS ID   ADV RTR   Seq NO Age Checksum
1 192.168.1.12 192.168.1.12 0x80000210 0 0xB196
```

Related Commands

prefix-list	Configures a prefix list to be used for OSPF routing.
route-map	Creates a route map for redistributing routes from one routing protocol to another. Used in configuring OSPF routing on the firewall.
router ospf	Configures global parameters for the OSPF routing processes on the firewall, and enables or disables OSPF routing through the firewall.
routing interface	Configures interface-specific OSPF routing parameters.

show ospf summary-address

Displays a list of all summary address redistribution information configured under an OSPF process.

show ospf summary-address

Syntax Description

summary-address	An address representing multiple (aggregated) addresses.

Defaults

None.

Command Modes

The **show** command is available in privileged mode.

Usage Guidelines

The OSPF routing-related **show** commands are available in privileged command mode on the firewall. You do not need to be in an OSPF configuration subcommand mode to use the OSPF-related **show** commands.

Examples

The following is sample output from the **show ospf summary-address** command for an OSPF process with the *pid* of 5:

```
pixfirewall# show ospf summary-address
OSPF Process 5, Summary-address
10.2.0.0/255.255.0.0 Metric -1, Type 0, Tag 0
10.2.0.0/255.255.0.0 Metric -1, Type 0, Tag 10
```

Related Commands

prefix-list	Configures a prefix list to be used for OSPF routing.
route-map	Creates a route map for redistributing routes from one routing protocol to another. Used in configuring OSPF routing on the firewall.
router ospf	Configures global parameters for the OSPF routing processes on the firewall, and enables or disables OSPF routing through the firewall.
routing interface	Configures interface-specific OSPF routing parameters.

show ospf virtual links

Displays parameters and the current state of OSPF virtual links.

show ospf virtual-links

Syntax Description

virtual-links	OSPF virtual links.

Defaults None.

Command Modes The **show ospf virtual-links** command is available in privileged mode.

Usage Guidelines The OSPF routing-related **show** commands are available in privileged command mode on the firewall. You do not need to be in an OSPF configuration subcommand mode to use the OSPF-related **show** commands.

Examples The following is sample output from the **show ospf virtual-links** command:

```
pixfirewall# show ospf virtual-links
Virtual Link to router 192.168.101.2 is up
Transit area 0.0.0.1, via interface Ethernet0, Cost of using 10
Transmit Delay is 1 sec, State POINT_TO_POINT
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 0:00:08
Adjacency State FULL
```

Related Commands

prefix-list	Configures a prefix list to be used for OSPF routing.
route-map	Creates a route map for redistributing routes from one routing protocol to another. Used in configuring OSPF routing on the firewall.
router ospf	Configures global parameters for the OSPF routing processes on the firewall, and enables or disables OSPF routing through the firewall.
routing interface	Configures interface-specific OSPF routing parameters.

show processes

Display processes.

show processes

Syntax Description

processes	The processes running on the PIX Firewall.

Command Modes Privileged mode.

Usage Guidelines The **show processes** command displays a list of the running processes. Processes are lightweight threads requiring only a few instructions. In the listing, PC is the program counter, SP is the stack pointer, STATE is the address of a thread queue, Runtime is the number of milliseconds that the thread has been running, SBASE is the stack base address, Stack is the current number of bytes used and the total size of the stack, and Process lists the thread's function.

Examples

The following is sample output from the **show processes** command:

```
pixfirewall(config)# show processes

    PC       SP       STATE       Runtime    SBASE     Stack Process
Hsi 001e7de9 0074e3ac 0054c8e0           0 0074d424 3884/4096 arp_timer
Lsi 001ecf55 007f15a4 0054c8e0          10 007f062c 3800/4096 FragDBGC
Lwe 00119af7 009bd7ec 00550040           0 009bc984 3688/4096 dbgtrace
Lwe 003da59d 009bf97c 00545218           0 009bda34 8008/8192 Logger
Hwe 003de658 009c2a74 005454c8           0 009c0afc 8024/8192 tcp_fast
Hwe 003de5d1 009c4b24 005454c8           0 009c2bac 8024/8192 tcp_slow
Lsi 002f8611 00af8e94 0054c8e0           0 00af7f0c 3944/4096 xlate clean
Lsi 002f851f 00af9f34 0054c8e0           0 00af8fbc 3884/4096 uxlate clean
Mwe 002ef7ff 00c6e304 0054c8e0           0 00c6c36c 7908/8192 tcp_intercept_times
Lsi 0042fb65 00d18b5c 0054c8e0           0 00d17bd4 3768/4096 route_process
Hsi 002e0b9c 00d19bec 0054c8e0          10 00d18c84 3780/4096 PIX Garbage Collecr
Hwe 00213ad9 00d2391c 0054c8e0           0 00d1f9b4 16048/16384 isakmp_time_keepr
Lsi 002de91c 00d3cc84 0054c8e0           0 00d3bcfc 3944/4096 perfmon
Mwe 0020b339 00d670b4 0054c8e0           0 00d6513c 7860/8192 IPsec timer handler
Hwe 00391143 00d7b9fc 005668f0           0 00d79ab4 6904/8192 qos_metric_daemon
Mwe 0025d205 00d92594 0054c8e0           0 00d91e2c 1436/2048 IP Background
Lwe 002f0302 00e44ee4 00561c08           0 00e4406c 3704/4096 pix/trace
Lwe 002f051e 00e45f94 00562338           0 00e4511c 3704/4096 pix/tconsole
H*  0011f4ef 0009fefc 0054c8c8        1580 00e4e484 13548/16384 ci/console
Csi 002e923b 00e5348c 0054c8e0           0 00e52534 3432/4096 update_cpu_usage
Hwe 002d63d1 00ef7324 0052bc98           0 00ef349c 15884/16384 uauth_in
Hwe 003dd0e5 00ef9424 00811bf8           0 00ef754c 7896/8192 uauth_thread
Hwe 003f2c62 00efa574 00545818           0 00ef95fc 3960/4096 udp_timer
Hsi 001dfcf2 00efc22c 0054c8e0           0 00efb2b4 3928/4096 557mcfix
Crd 001dfca7 00efd2ec 0054cd58   764174020 00efc364 3688/4096 557poll
Lsi 001dfd5d 00efe38c 0054c8e0           0 00efd414 3700/4096 557timer
Cwe 001e1785 00f1440c 0085b790         770 00f12514 7344/8192 pix/intf0
Mwe 003f29d2 00f154fc 0085a420           0 00f145c4 3896/4096 riprx/0
Msi 0039a3a1 00f1660c 0054c8e0           0 00f15694 3888/4096 riptx/0
Cwe 001e1785 00f1c744 008d0d00           0 00f1a84c 7928/8192 pix/intf1
Mwe 003f29d2 00f1d854 0085a3d8           0 00f1c91c 3896/4096 riprx/1
Msi 0039a3a1 00f1e964 0054c8e0           0 00f1d9ec 3888/4096 riptx/1
Cwe 001ea085 00f24b0c 0071aa6c           0 00f22ba4 8040/8192 pix/intf2
Mwe 003f29d2 00f25bac 0085a390           0 00f24c74 3896/4096 riprx/2
Msi 0039a3a1 00f26cbc 0054c8e0           0 00f25d44 3888/4096 riptx/2
Hwe 003dd379 00f4c3b4 007fd000           0 00f4c10c  300/1024 listen/http1
Mwe 00367556 00f4e60c 0054c8e0           0 00f4c694 7640/8192 Crypto CA
Mrd 002650c9 00f7bf3c 0054c918        4780 00f79fc4 7744/8192 OSPF Router
Mrd 00265869 00f7960c 0054c918        4760 00f78ed4 1608/2048 OSPF Hello
```

show routing

Displays the (non-default) interface-specific routing configuration.

show routing [**interface** *if_name*]

Syntax Description

if_name	The name of the interface for which to display the configuration.

Defaults

None.

Command Modes The **show routing** command is available in privileged mode.

Usage Guidelines The OSPF routing-related **show** commands are available in privileged command mode on the firewall. You do not need to be in an OSPF configuration subcommand mode to use the OSPF-related **show** commands.

Examples The following is sample output from the **show routing** command:

```
pixfirewall# show routing
routing interface outside
  ospf retransmit-interval 15
routing interface inside
  ospf cost 206
```

The following is sample output from the **show routing** [**interface** *if_name*] command:

```
pixfirewall# show routing interface outside
routing interface outside
  ospf retransmit-interval 15
```

Related Commands

prefix-list	Configures a prefix list to be used for OSPF routing.
route-map	Creates a route map for redistributing routes from one routing protocol to another. Used in configuring OSPF routing on the firewall.
router ospf	Configures global parameters for the OSPF routing processes on the firewall, and enables or disables OSPF routing through the firewall.
routing interface	Configures interface-specific OSPF routing parameters.

show running-config

Display the PIX Firewall running configuration.

show running-config

Syntax Description

running-config	The configuration running on the PIX Firewall.

Command Modes Privileged mode.

Usage Guidelines The **show running-config** command displays the current running configuration. The keyword **running-config** is used to match the Cisco IOS software command. The **show running-config** command output is the same as the pre-existing PIX Firewall **write terminal** command.

The **running-config** keyword can be used only in the **show running-config** command. It cannot be used with **no** or **clear**, or as a standalone command. If it is, the CLI treats it as a non-supported command. Also, for this reason, when **?**, **no ?**, or **clear ?** are entered, a **running-config** option is not listed in the command list.

Note PIX Device Manager (PDM) commands will appear in your configuration after you use PDM to connect to or configure your PIX Firewall.

Examples

The following is sample output from the **show running-config** command:

```
pixfirewall# show running-config
: Saved
:
PIX Version 6.2(1)
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 intf2 security10
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname pixdoc515
domain-name cisco.com
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol ils 389
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol smtp 25
fixup protocol snmp 161-162
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
names
access-list inside_outbound_nat0_acl permit ip 10.1.3.0 255.255.255.0 10.1.2.0
access-list inside_outbound_nat0_acl permit ip any any
access-list outside_cryptomap_20 permit ip 10.1.3.0 255.255.255.0 10.1.2.0 255.
access-list outside_cryptomap_40 permit ip any any
access-list 101 permit ip any any
pager lines 24
logging on
interface ethernet0 10baset
interface ethernet1 100full
interface ethernet2 100full shutdown
icmp permit any outside
icmp permit any inside
mtu outside 1500
mtu inside 1500
mtu intf2 1500
ip address outside 172.23.59.230 255.255.0.0 pppoe
ip address inside 10.1.3.1 255.255.255.0
ip address intf2 127.0.0.1 255.255.255.0
multicast interface inside
ip audit info action alarm
ip audit attack action alarm
no failover
failover timeout 0:00:00
failover poll 15
failover ip address outside 0.0.0.0
```

```
failover ip address inside 0.0.0.0
failover ip address intf2 0.0.0.0
pdm location 10.1.2.1 255.255.255.255 outside
pdm location 10.1.2.0 255.255.255.0 outside
pdm logging alerts 100
pdm history enable
arp timeout 14400
global (inside) 6 192.168.1.2-192.168.1.3
global (inside) 3 192.168.4.1
nat (inside) 0 access-list inside_outbound_nat0_acl
access-group 101 in interface outside
route outside 0.0.0.0 0.0.0.0 172.23.59.225 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h323 0:05:00 s0
timeout uauth 0:05:00 absolute
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
aaa-server LOCAL protocol local
http server enable
http 0.0.0.0 0.0.0.0 outside
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
floodguard enable
sysopt connection permit-ipsec
crypto ipsec transform-set ESP-DES-SHA esp-des esp-sha-hmac
crypto map outside_map 20 ipsec-isakmp
crypto map outside_map 20 match address outside_cryptomap_20
crypto map outside_map 20 set peer 172.23.59.231
crypto map outside_map 20 set transform-set ESP-DES-SHA
crypto map outside_map 40 ipsec-isakmp
crypto map outside_map 40 match address outside_cryptomap_40
crypto map outside_map 40 set peer 123.5.5.5
isakmp key ******** address 172.23.59.231 netmask 255.255.255.255 no-xauth no-c
isakmp peer fqdn no-xauth no-config-mode
isakmp policy 20 authentication pre-share
isakmp policy 20 encryption des
isakmp policy 20 hash sha
isakmp policy 20 group 2
isakmp policy 20 lifetime 86400
isakmp policy 40 authentication rsa-sig
isakmp policy 40 encryption 3des
isakmp policy 40 hash sha
isakmp policy 40 group 2
isakmp policy 40 lifetime 86400
telnet timeout 5
ssh timeout 5
console timeout 10
dhcprelay timeout 60
terminal width 80
Cryptochecksum:4d600490f46b5d335c0fbf2eda0015a2
: end
```

Note A configuration error at bootup will cause the cryptochecksum to display all zeros. Perform the **write memory** command, then the **show running-config** command again to diplay the proper checksum.

show startup-config

Display the PIX Firewall startup configuration.

show startup-config

Syntax Description

startup-config	The configuration present at startup on the PIX Firewall.

Command Modes

Privileged mode.

Usage Guidelines

The **show startup-config** command displays the startup configuration of the PIX Firewall. The keyword **startup-config** is used to match the Cisco IOS software command. The **show startup-config** command output is the same as the pre-existing PIX Firewall **show configure** command. The **show startup-config** command is not needed for PDM but is provided for compatibility with Cisco IOS software.

The **startup-config** keyword can be used only in the **show startup-config** command. It cannot be used with **no** or **clear**, or as a standalone command. If it is, the CLI treats it as a non-supported command. Also, for this reason, when **?**, **no ?**, or **clear ?** are entered, a **startup-config** option is not listed in the command list.

Examples

The following is sample output from the **show startup-config** command:

```
pixfirewall# show startup-config
: Saved
: Written by enable_15 at 17:14:09.092 UTC Tue Apr 9 2002
PIX Version 6.2(0)227
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 intf2 security10
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname pixdoc515
domain-name cisco.com
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol ils 389
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
names
access-list inside_outbound_nat0_acl permit ip 10.1.3.0 255.255.255.0 10.1.2.0
access-list inside_outbound_nat0_acl permit ip any any
access-list outside_cryptomap_20 permit ip 10.1.3.0 255.255.255.0 10.1.2.0 255.
access-list outside_cryptomap_40 permit ip any any
access-list 101 permit ip any any
pager lines 24
logging on
interface ethernet0 10baset
```

```
interface ethernet1 100full
interface ethernet2 100full shutdown
icmp permit any outside
icmp permit any inside
mtu outside 1500
mtu inside 1500
mtu intf2 1500
ip address outside 172.23.59.230 255.255.0.0 pppoe
ip address inside 10.1.3.1 255.255.255.0
ip address intf2 127.0.0.1 255.255.255.0
multicast interface inside
ip audit info action alarm
ip audit attack action alarm
no failover
failover timeout 0:00:00
failover poll 15
failover ip address outside 0.0.0.0
failover ip address inside 0.0.0.0
failover ip address intf2 0.0.0.0
pdm location 10.1.2.1 255.255.255.255 outside
pdm location 10.1.2.0 255.255.255.0 outside
pdm logging alerts 100
pdm history enable
arp timeout 14400
global (inside) 6 192.168.1.2-192.168.1.3
global (inside) 3 192.168.4.1
nat (inside) 0 access-list inside_outbound_nat0_acl
access-group 101 in interface outside
route outside 0.0.0.0 0.0.0.0 172.23.59.225 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h323 0:05:00 s0
timeout uauth 0:05:00 absolute
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
aaa-server LOCAL protocol local
http server enable
http 0.0.0.0 0.0.0.0 outside
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
floodguard enable
sysopt connection permit-ipsec
crypto ipsec transform-set ESP-DES-SHA esp-des esp-sha-hmac
crypto map outside_map 20 ipsec-isakmp
crypto map outside_map 20 match address outside_cryptomap_20
crypto map outside_map 20 set peer 172.23.59.231
crypto map outside_map 20 set transform-set ESP-DES-SHA
crypto map outside_map 40 ipsec-isakmp
crypto map outside_map 40 match address outside_cryptomap_40
crypto map outside_map 40 set peer 123.5.5.5
isakmp key ******** address 172.23.59.231 netmask 255.255.255.255 no-xauth no-c
isakmp peer fqdn no-xauth no-config-mode
isakmp policy 20 authentication pre-share
isakmp policy 20 encryption des
isakmp policy 20 hash sha
isakmp policy 20 group 2
isakmp policy 20 lifetime 86400
isakmp policy 40 authentication rsa-sig
isakmp policy 40 encryption 3des
isakmp policy 40 hash sha
isakmp policy 40 group 2
isakmp policy 40 lifetime 86400
telnet timeout 5
```

```
ssh timeout 5
```

show tech-support

View information to help a support analyst.

show tech-support [**no-config**]

Syntax Description

no-config	Excludes the output of the running configuration.
tech-support	The data used for diagnosis by technical support analysts.

Command Modes

Privileged mode.

Usage Guidelines

The **show tech-suppor**t command lists information that technical support analysts need to help you diagnose PIX Firewall problems. This command combines the output from the **show** commands that provide the most information to a technical support analyst.

Examples

The following is sample output from the **show tech-support no-config** command, which excludes the running configuration:

```
pixfirewall(config)# show tech-support no-config

Cisco PIX Firewall Version 6.3(1)
Cisco PIX Device Manager Version 2.1(1)

Compiled on Fri 15-Nov-02 14:35 by root

pixfirewall up 2 days 8 hours

Hardware:   PIX-515, 64 MB RAM, CPU Pentium 200 MHz
Flash i28F640J5 @ 0x300, 16MB
BIOS Flash AT29C257 @ 0xfffd8000, 32KB

0: ethernet0: address is 0003.e300.73fd, irq 10
1: ethernet1: address is 0003.e300.73fe, irq 7
2: ethernet2: address is 00d0.b7c8.139e, irq 9
Licensed Features:
Failover:           Disabled
VPN-DES:            Enabled
VPN-3DES-AES:       Disabled
Maximum Interfaces: 3
Cut-through Proxy:  Enabled
Guards:             Enabled
URL-filtering:      Enabled
Inside Hosts:       Unlimited
Throughput:         Unlimited
IKE peers:          Unlimited

This PIX has a Restricted (R) license.

Serial Number: 480430455 (0x1ca2c977)
Running Activation Key: 0xc2e94182 0xc21d8206 0x15353200 0x633f6734
Configuration last modified by enable_15 at 23:05:24.264 UTC Sat Nov 16 2002
```

```
------------------ show clock ------------------

00:08:14.911 UTC Sun Nov 17 2002

------------------ show memory ------------------

Free memory:        50708168 bytes
Used memory:        16400696 bytes
-------------     ----------------
Total memory:       67108864 bytes

------------------ show conn count ------------------

0 in use, 0 most used

------------------ show xlate count ------------------

0 in use, 0 most used

------------------ show blocks ------------------

  SIZE    MAX    LOW    CNT
     4   1600   1600   1600
    80    400    400    400
   256    500    499    500
  1550   1188    795    919

------------------ show interface ------------------

interface ethernet0 "outside" is up, line protocol is up
  Hardware is i82559 ethernet, address is 0003.e300.73fd
  IP address 172.23.59.232, subnet mask 255.255.0.0
  MTU 1500 bytes, BW 10000 Kbit half duplex
        1267 packets input, 185042 bytes, 0 no buffer
        Received 1248 broadcasts, 0 runts, 0 giants
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
        20 packets output, 1352 bytes, 0 underruns
        0 output errors, 0 collisions, 0 interface resets
        0 babbles, 0 late collisions, 9 deferred
        0 lost carrier, 0 no carrier
        input queue (curr/max blocks): hardware (13/128) software (0/2)
        output queue (curr/max blocks): hardware (0/1) software (0/1)
interface ethernet1 "inside" is up, line protocol is down
  Hardware is i82559 ethernet, address is 0003.e300.73fe
  IP address 10.1.1.1, subnet mask 255.255.255.0
  MTU 1500 bytes, BW 10000 Kbit half duplex
        0 packets input, 0 bytes, 0 no buffer
        Received 0 broadcasts, 0 runts, 0 giants
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
        1 packets output, 60 bytes, 0 underruns
        0 output errors, 0 collisions, 0 interface resets
        0 babbles, 0 late collisions, 0 deferred
        1 lost carrier, 0 no carrier
        input queue (curr/max blocks): hardware (128/128) software (0/0)
        output queue (curr/max blocks): hardware (0/1) software (0/1)
interface ethernet2 "intf2" is administratively down, line protocol is down
  Hardware is i82559 ethernet, address is 00d0.b7c8.139e
  IP address 127.0.0.1, subnet mask 255.255.255.255
  MTU 1500 bytes, BW 10000 Kbit half duplex
        0 packets input, 0 bytes, 0 no buffer
        Received 0 broadcasts, 0 runts, 0 giants
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
        0 packets output, 0 bytes, 0 underruns
        0 output errors, 0 collisions, 0 interface resets
```

```
        0 babbles, 0 late collisions, 0 deferred
        0 lost carrier, 0 no carrier
        input queue (curr/max blocks): hardware (128/128) software (0/0)
        output queue (curr/max blocks): hardware (0/0) software (0/0)

------------------ show cpu usage ------------------

CPU utilization for 5 seconds = 0%; 1 minute: 0%; 5 minutes: 0%

------------------ show process ------------------


    PC       SP       STATE       Runtime    SBASE     Stack Process
Hsi 001e3329 00763e7c 0053e5c8           0 00762ef4 3784/4096 arp_timer
Lsi 001e80e9 00807074 0053e5c8           0 008060fc 3832/4096 FragDBGC
Lwe 00117e3a 009dc2e4 00541d18           0 009db46c 3704/4096 dbgtrace
Lwe 003cee95 009de464 00537718           0 009dc51c 8008/8192 Logger
Hwe 003d2d18 009e155c 005379c8           0 009df5e4 8008/8192 tcp_fast
Hwe 003d2c91 009e360c 005379c8           0 009e1694 8008/8192 tcp_slow
Lsi 002ec97d 00b1a464 0053e5c8           0 00b194dc 3928/4096 xlate clean
Lsi 002ec88b 00b1b504 0053e5c8           0 00b1a58c 3888/4096 uxlate clean
Mwe 002e3a17 00c8f8d4 0053e5c8           0 00c8d93c 7908/8192 tcp_intercept_times
Lsi 00423dd5 00d3a22c 0053e5c8           0 00d392a4 3900/4096 route_process
Hsi 002d59fc 00d3b2bc 0053e5c8           0 00d3a354 3780/4096 PIX Garbage Collecr
Hwe 0020e301 00d5957c 0053e5c8           0 00d55614 16048/16384 isakmp_time_keepr
Lsi 002d377c 00d7292c 0053e5c8           0 00d719a4 3928/4096 perfmon
Hwe 0020bd07 00d9c12c 0050bb90           0 00d9b1c4 3944/4096 IPSec
Mwe 00205e25 00d9e1ec 0053e5c8           0 00d9c274 7860/8192 IPsec timer handler
Hwe 003864e3 00db26bc 00557920           0 00db0764 6952/8192 qos_metric_daemon
Mwe 00255a65 00dc9244 0053e5c8           0 00dc8adc 1436/2048 IP Background
Lwe 002e450e 00e7bb94 00552c30           0 00e7ad1c 3704/4096 pix/trace
Lwe 002e471e 00e7cc44 00553368           0 00e7bdcc 3704/4096 pix/tconsole
Hwe 001e5368 00e7ed44 00730674           0 00e7ce9c 7228/8192 pix/intf0
Hwe 001e5368 00e80e14 007305d4           0 00e7ef6c 7228/8192 pix/intf1
Hwe 001e5368 00e82ee4 00730534        2470 00e8103c 4892/8192 pix/intf2
H*  0011d7f7 0009ff2c 0053e5b0         780 00e8511c 13004/16384 ci/console
Csi 002dd8ab 00e8a124 0053e5c8           0 00e891cc 3396/4096 update_cpu_usage
Hwe 002cb4d1 00f2bfbc 0051e360           0 00f2a134 7692/8192 uauth_in
Hwe 003d17d1 00f2e0bc 00828cf0           0 00f2c1e4 7896/8192 uauth_thread
Hwe 003e71d4 00f2f20c 00537d20           0 00f2e294 3960/4096 udp_timer
Hsi 001db3ca 00f30fc4 0053e5c8           0 00f3004c 3784/4096 557mcfix
Crd 001db37f 00f32084 0053ea40   121094970 00f310fc 3744/4096 557poll
Lsi 001db435 00f33124 0053e5c8           0 00f321ac 3700/4096 557timer
Hwe 001e5398 00f441dc 008121e0           0 00f43294 3912/4096 fover_ip0
Cwe 001dcdad 00f4523c 00872b48          20 00f44344 3528/4096 ip/0:0
Hwe 001e5398 00f4633c 008121bc           0 00f453f4 3532/4096 icmp0
Hwe 001e5398 00f47404 00812198           0 00f464cc 3896/4096 udp_thread/0
Hwe 001e5398 00f4849c 00812174           0 00f475a4 3832/4096 tcp_thread/0
Hwe 001e5398 00f495bc 00812150           0 00f48674 3912/4096 fover_ip1
Cwe 001dcdad 00f4a61c 008ea850           0 00f49724 3832/4096 ip/1:1
Hwe 001e5398 00f4b71c 0081212c           0 00f4a7d4 3912/4096 icmp1
Hwe 001e5398 00f4c7e4 00812108           0 00f4b8ac 3896/4096 udp_thread/1
Hwe 001e5398 00f4d87c 008120e4           0 00f4c984 3832/4096 tcp_thread/1
Hwe 001e5398 00f4e99c 008120c0           0 00f4da54 3912/4096 fover_ip2
Cwe 001e542d 00f4fa6c 00730534           0 00f4eb04 3944/4096 ip/2:2
Hwe 001e5398 00f50afc 0081209c           0 00f4fbb4 3912/4096 icmp2
Hwe 001e5398 00f51bc4 00812078           0 00f50c8c 3896/4096 udp_thread/2
Hwe 001e5398 00f52c5c 00812054           0 00f51d64 3832/4096 tcp_thread/2
Hwe 003d1a65 00f78284 008140f8           0 00f77fdc  300/1024 listen/http1
Mwe 0035cafa 00f7a63c 0053e5c8           0 00f786c4 7640/8192 Crypto CA

------------------ show failover ------------------

No license for Failover
```

```
------------------ show traffic ------------------

outside:
        received (in 205213.390 secs):
                1267 packets    185042 bytes
                0 pkts/sec      0 bytes/sec
        transmitted (in 205213.390 secs):
                20 packets      1352 bytes
                0 pkts/sec      0 bytes/sec
inside:
        received (in 205215.800 secs):
                0 packets       0 bytes
                0 pkts/sec      0 bytes/sec
        transmitted (in 205215.800 secs):
                1 packets       60 bytes
                0 pkts/sec      0 bytes/sec
intf2:
        received (in 205215.810 secs):
                0 packets       0 bytes
                0 pkts/sec      0 bytes/sec
        transmitted (in 205215.810 secs):
                0 packets       0 bytes
                0 pkts/sec      0 bytes/sec

------------------ show perfmon ------------------


PERFMON STATS:    Current      Average
Xlates               0/s          0/s
Connections          0/s          0/s
TCP Conns            0/s          0/s
UDP Conns            0/s          0/s
URL Access           0/s          0/s
URL Server Req       0/s          0/s
TCP Fixup            0/s          0/s
TCPIntercept         0/s          0/s
HTTP Fixup           0/s          0/s
FTP Fixup            0/s          0/s
AAA Authen           0/s          0/s
AAA Author           0/s          0/s
AAA Account          0/s          0/s
```

The following is sample output from the **show tech-support** command, which includes the running configuration:

```
pixfirewall(config)# show tech-support

Cisco PIX Firewall Version 6.3(1)
Cisco PIX Device Manager Version 2.1(1)

Compiled on Fri 15-Nov-02 14:35 by root

pixfirewall up 2 days 9 hours

Hardware:   PIX-515, 64 MB RAM, CPU Pentium 200 MHz
Flash i28F640J5 @ 0x300, 16MB
BIOS Flash AT29C257 @ 0xfffd8000, 32KB

0: ethernet0: address is 0003.e300.73fd, irq 10
1: ethernet1: address is 0003.e300.73fe, irq 7
2: ethernet2: address is 00d0.b7c8.139e, irq 9
Licensed Features:
Failover:           Disabled
```

```
VPN-DES:            Enabled
VPN-3DES-AES:       Disabled
Maximum Interfaces: 3
Cut-through Proxy:  Enabled
Guards:             Enabled
URL-filtering:      Enabled
Inside Hosts:       Unlimited
Throughput:         Unlimited
IKE peers:          Unlimited

This PIX has a Restricted (R) license.

Serial Number: 480430455 (0x1ca2c977)
Running Activation Key: 0xc2e94182 0xc21d8206 0x15353200 0x633f6734
Configuration last modified by enable_15 at 23:05:24.264 UTC Sat Nov 16 2002

------------------ show clock ------------------

00:08:39.591 UTC Sun Nov 17 2002

------------------ show memory ------------------

Free memory:        50708168 bytes
Used memory:        16400696 bytes
-------------     ----------------
Total memory:       67108864 bytes

------------------ show conn count ------------------

0 in use, 0 most used

------------------ show xlate count ------------------

0 in use, 0 most used

------------------ show blocks ------------------

  SIZE    MAX    LOW    CNT
     4   1600   1600   1600
    80    400    400    400
   256    500    499    500
  1550   1188    795    919

------------------ show interface ------------------

interface ethernet0 "outside" is up, line protocol is up
  Hardware is i82559 ethernet, address is 0003.e300.73fd
  IP address 172.23.59.232, subnet mask 255.255.0.0
  MTU 1500 bytes, BW 10000 Kbit half duplex
        1267 packets input, 185042 bytes, 0 no buffer
        Received 1248 broadcasts, 0 runts, 0 giants
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
        20 packets output, 1352 bytes, 0 underruns
        0 output errors, 0 collisions, 0 interface resets
        0 babbles, 0 late collisions, 9 deferred
        0 lost carrier, 0 no carrier
        input queue (curr/max blocks): hardware (13/128) software (0/2)
        output queue (curr/max blocks): hardware (0/1) software (0/1)
interface ethernet1 "inside" is up, line protocol is down
  Hardware is i82559 ethernet, address is 0003.e300.73fe
  IP address 10.1.1.1, subnet mask 255.255.255.0
  MTU 1500 bytes, BW 10000 Kbit half duplex
        0 packets input, 0 bytes, 0 no buffer
        Received 0 broadcasts, 0 runts, 0 giants
```

```
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
        1 packets output, 60 bytes, 0 underruns
        0 output errors, 0 collisions, 0 interface resets
        0 babbles, 0 late collisions, 0 deferred
        1 lost carrier, 0 no carrier
        input queue (curr/max blocks): hardware (128/128) software (0/0)
        output queue (curr/max blocks): hardware (0/1) software (0/1)
interface ethernet2 "intf2" is administratively down, line protocol is down
  Hardware is i82559 ethernet, address is 00d0.b7c8.139e
  IP address 127.0.0.1, subnet mask 255.255.255.255
  MTU 1500 bytes, BW 10000 Kbit half duplex
        0 packets input, 0 bytes, 0 no buffer
        Received 0 broadcasts, 0 runts, 0 giants
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
        0 packets output, 0 bytes, 0 underruns
        0 output errors, 0 collisions, 0 interface resets
        0 babbles, 0 late collisions, 0 deferred
        0 lost carrier, 0 no carrier
        input queue (curr/max blocks): hardware (128/128) software (0/0)
        output queue (curr/max blocks): hardware (0/0) software (0/0)

------------------ show cpu usage ------------------

CPU utilization for 5 seconds = 0%; 1 minute: 0%; 5 minutes: 0%

------------------ show process ------------------


    PC       SP       STATE       Runtime    SBASE     Stack Process
Hsi 001e3329 00763e7c 0053e5c8          0 00762ef4 3784/4096 arp_timer
Lsi 001e80e9 00807074 0053e5c8          0 008060fc 3832/4096 FragDBGC
Lwe 00117e3a 009dc2e4 00541d18          0 009db46c 3704/4096 dbgtrace
Lwe 003cee95 009de464 00537718          0 009dc51c 8008/8192 Logger
Hwe 003d2d18 009e155c 005379c8          0 009df5e4 8008/8192 tcp_fast
Hwe 003d2c91 009e360c 005379c8          0 009e1694 8008/8192 tcp_slow
Lsi 002ec97d 00b1a464 0053e5c8          0 00b194dc 3928/4096 xlate clean
Lsi 002ec88b 00b1b504 0053e5c8          0 00b1a58c 3888/4096 uxlate clean
Mwe 002e3a17 00c8f8d4 0053e5c8          0 00c8d93c 7908/8192 tcp_intercept_times
Lsi 00423dd5 00d3a22c 0053e5c8          0 00d392a4 3900/4096 route_process
Hsi 002d59fc 00d3b2bc 0053e5c8          0 00d3a354 3780/4096 PIX Garbage Collecr
Hwe 0020e301 00d5957c 0053e5c8          0 00d55614 16048/16384 isakmp_time_keepr
Lsi 002d377c 00d7292c 0053e5c8          0 00d719a4 3928/4096 perfmon
Hwe 0020bd07 00d9c12c 0050bb90          0 00d9b1c4 3944/4096 IPSec
Mwe 00205e25 00d9e1ec 0053e5c8          0 00d9c274 7860/8192 IPsec timer handler
Hwe 003864e3 00db26bc 00557920          0 00db0764 6952/8192 qos_metric_daemon
Mwe 00255a65 00dc9244 0053e5c8          0 00dc8adc 1436/2048 IP Background
Lwe 002e450e 00e7bb94 00552c30          0 00e7ad1c 3704/4096 pix/trace
Lwe 002e471e 00e7cc44 00553368          0 00e7bdcc 3704/4096 pix/tconsole
Hwe 001e5368 00e7ed44 00730674          0 00e7ce9c 7228/8192 pix/intf0
Hwe 001e5368 00e80e14 007305d4          0 00e7ef6c 7228/8192 pix/intf1
Hwe 001e5368 00e82ee4 00730534       2470 00e8103c 4892/8192 pix/intf2
H*  0011d7f7 0009ff2c 0053e5b0        950 00e8511c 13004/16384 ci/console
Csi 002dd8ab 00e8a124 0053e5c8          0 00e891cc 3396/4096 update_cpu_usage
Hwe 002cb4d1 00f2bfbc 0051e360          0 00f2a134 7692/8192 uauth_in
Hwe 003d17d1 00f2e0bc 00828cf0          0 00f2c1e4 7896/8192 uauth_thread
Hwe 003e71d4 00f2f20c 00537d20          0 00f2e294 3960/4096 udp_timer
Hsi 001db3ca 00f30fc4 0053e5c8          0 00f3004c 3784/4096 557mcfix
Crd 001db37f 00f32084 0053ea40  121109610 00f310fc 3744/4096 557poll
Lsi 001db435 00f33124 0053e5c8          0 00f321ac 3700/4096 557timer
Hwe 001e5398 00f441dc 008121e0          0 00f43294 3912/4096 fover_ip0
Cwe 001dcdad 00f4523c 00872b48         20 00f44344 3528/4096 ip/0:0
Hwe 001e5398 00f4633c 008121bc          0 00f453f4 3532/4096 icmp0
Hwe 001e5398 00f47404 00812198          0 00f464cc 3896/4096 udp_thread/0
Hwe 001e5398 00f4849c 00812174          0 00f475a4 3832/4096 tcp_thread/0
```

```
Hwe 001e5398 00f495bc 00812150          0 00f48674 3912/4096 fover_ip1
Cwe 001dcdad 00f4a61c 008ea850          0 00f49724 3832/4096 ip/1:1
Hwe 001e5398 00f4b71c 0081212c          0 00f4a7d4 3912/4096 icmp1
Hwe 001e5398 00f4c7e4 00812108          0 00f4b8ac 3896/4096 udp_thread/1
Hwe 001e5398 00f4d87c 008120e4          0 00f4c984 3832/4096 tcp_thread/1
Hwe 001e5398 00f4e99c 008120c0          0 00f4da54 3912/4096 fover_ip2
Cwe 001e542d 00f4fa6c 00730534          0 00f4eb04 3944/4096 ip/2:2
Hwe 001e5398 00f50afc 0081209c          0 00f4fbb4 3912/4096 icmp2
Hwe 001e5398 00f51bc4 00812078          0 00f50c8c 3896/4096 udp_thread/2
Hwe 001e5398 00f52c5c 00812054          0 00f51d64 3832/4096 tcp_thread/2
Hwe 003d1a65 00f78284 008140f8          0 00f77fdc  300/1024 listen/http1
Mwe 0035cafa 00f7a63c 0053e5c8          0 00f786c4 7640/8192 Crypto CA

------------------ show failover ------------------

No license for Failover

------------------ show traffic ------------------

outside:
        received (in 205238.740 secs):
                1267 packets    185042 bytes
                0 pkts/sec      0 bytes/sec
        transmitted (in 205238.740 secs):
                20 packets      1352 bytes
                0 pkts/sec      0 bytes/sec
inside:
        received (in 205242.200 secs):
                0 packets       0 bytes
                0 pkts/sec      0 bytes/sec
        transmitted (in 205242.200 secs):
                1 packets       60 bytes
                0 pkts/sec      0 bytes/sec
intf2:
        received (in 205242.200 secs):
                0 packets       0 bytes
                0 pkts/sec      0 bytes/sec
        transmitted (in 205242.200 secs):
                0 packets       0 bytes
                0 pkts/sec      0 bytes/sec

------------------ show perfmon ------------------


PERFMON STATS:    Current      Average
Xlates               0/s          0/s
Connections          0/s          0/s
TCP Conns            0/s          0/s
UDP Conns            0/s          0/s
URL Access           0/s          0/s
URL Server Req       0/s          0/s
TCP Fixup            0/s          0/s
TCPIntercept         0/s          0/s
HTTP Fixup           0/s          0/s
FTP Fixup            0/s          0/s
AAA Authen           0/s          0/s
AAA Author           0/s          0/s
AAA Account          0/s          0/s

------------------ show running-config ------------------

: Saved
:
PIX Version 6.3(1)
```

```
interface ethernet0 auto
interface ethernet1 auto
interface ethernet2 auto shutdown
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 intf2 security10
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname pixfirewall
domain-name cisco.com
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol ils 389
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
fixup protocol sip udp 5060
names
access-list 101 permit tcp any host 10.1.1.3 eq www
access-list 101 permit tcp any host 10.1.1.3 eq smtp
pager lines 24
mtu outside 1500
mtu inside 1500
mtu intf2 1500
ip address outside 172.23.59.232 255.255.0.0
ip address inside 10.1.1.1 255.255.255.0
ip address intf2 127.0.0.1 255.255.255.255
ip audit info action alarm
ip audit attack action alarm
pdm history enable
arp timeout 14400
global (outside) 1 interface
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
route-map maptag1 permit 8
  set metric 5
  set metric-type type-2
  match metric 5
route outside 0.0.0.0 0.0.0.0 172.23.59.225 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h225 1:00:00
timeout h323 0:05:00 mgcp 0:05:00 sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
aaa-server LOCAL protocol local
http server enable
http 10.1.1.2 255.255.255.255 inside
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
floodguard enable
telnet timeout 5
ssh timeout 5
console timeout 0
terminal width 80
banner exec working...
banner motd Haveagoodday
```

```
Cryptochecksum:00000000000000000000000000000000
: end
```

show tcpstat

Displays the status of the firewall TCP stack and the TCP connections terminated on the firewall (for debugging).

show tcpstat

Syntax Description

tcpstat	TCP connection statistics.

Defaults

None.

Command Modes

The **show tcpstat** command is available in privileged mode.

Usage Guidelines

The **show tcpstat** command displays the status of the TCP stack and TCP connections terminated on the firewall. The TCP statistics displayed are described in Table 8-4:

Table 8-4 TCP Statistics in the show tcpstat Command

Statistic	Description
tcb_cnt	The number of TCP users.
proxy_cnt	The number of TCP proxies. TCP proxies are used by user authorization.
tcp_xmt pkts	The number of packets that were transmitted by the TCP stack.
tcp_rcv good pkts	The number of good packets that were received by the TCP stack.
tcp_rcv drop pkts	The number of received packets that the TCP stack dropped.
tcp bad chksum	The number of received packets that had a bad checksum.
tcp user hash add	The number of TCP users that were added to the hash table.
tcp user hash add dup	The number of times a TCP user was already in the hash table when trying to add a new user.
tcp user srch hash hit	The number of times a TCP user was found in the hash table when searching.
tcp user srch hash miss	The number of times a TCP user was not found in the hash table when searching.
tcp user hash delete	The number of times a TCP user was deleted from the hash table.

Table 8-4 TCP Statistics in the show tcpstat Command (continued)

Statistic	Description
tcp user hash delete miss	The number of times a TCP user was not found in the hash table when trying to delete the user.
lip	The local IP address of the TCP user.
fip	The foreign IP address of the TCP user.
lp	The local port of the TCP user.
fp	The foreign port of the TCP user.
st	The state (see RFC 793) of the TCP user. The possible values are as follows: 1 CLOSED 2 LISTEN 3 SYN_SENT 4 SYN_RCVD 5 ESTABLISHED 6 FIN_WAIT_1 7 FIN_WAIT_2 8 CLOSE_WAIT 9 CLOSING 10 LAST_ACK 11 TIME_WAIT
rexqlen	The length of the retransmit queue of the TCP user.
inqlen	The length of the input queue of the TCP user.
tw_timer	The value of the time_wait timer (in milliseconds) of the TCP user.
to_timer	The value of the inactivity timeout timer (in milliseconds) of the TCP user.
cl_timer	The value of the close request timer (in milliseconds) of the TCP user.
per_timer	The value of the persist timer (in milliseconds) of the TCP user.
rt_timer	The value of the retransmit timer (in milliseconds) of the TCP user.
tries	The retransmit count of the TCP user.

Examples

The following example shows the output from the **show tcpstat** command:

```
pixfirewall(config)# show tcpstat
                 CURRENT MAX     TOTAL
tcb_cnt          2       12      320
proxy_cnt        0       0       160

tcp_xmt pkts = 540591
tcp_rcv good pkts = 6583
tcp_rcv drop pkts = 2
tcp bad chksum = 0
tcp user hash add = 2028
tcp user hash add dup = 0
tcp user srch hash hit = 316753
```

```
tcp user srch hash miss = 6663
tcp user hash delete = 2027
tcp user hash delete miss = 0

lip = 172.23.59.230 fip = 10.21.96.254 lp = 443 fp = 2567 st
= 4 rexqlen = 0
in0
  tw_timer = 0 to_timer = 179000 cl_timer = 0 per_timer = 0
rt_timer = 0
tries 0
```

Related Commands

show conn	Displays all active connections.

show traffic/clear traffic

Shows interface transmit and receive activity.

clear traffic

show traffic

Syntax Description

traffic	The packets and bytes moving through an interface.

Command Modes

Privileged mode.

Usage Guidelines

The **show traffic** command lists the number of packets and bytes moving through each interface. The number of seconds is the duration the PIX Firewall has been online since the last reboot. The **clear traffic** command clears counters for the **show traffic** command output.

Examples

The following is sample output from the **show traffic** command:

```
show traffic
outside:
        received (in 3786 secs):
                97 packets      6191 bytes
                42 pkts/sec     1 bytes/sec
        transmitted (in 3786 secs):
                99 packets      10590 bytes
                0 pkts/sec      2 bytes/sec ...
```

show uauth/clear uauth

Display or delete all authorization caches for a user.

clear uauth [*username*]

show uauth [*username*]

Syntax Description

username	Clear or view user authentication information by username.

Command Modes

Privileged mode.

Usage Guidelines

The **show uauth** command displays one or all currently authenticated users, the host IP to which they are bound, and, if applicable, any cached IP and port authorization information.

The **clear uauth** command deletes one user's, or all users, AAA authorization and authentication caches, which forces the user or users to reauthenticate the next time they create a connection. The **show uauth** command also lists CiscoSecure 2.1 and later idletime and timeout values, which can be set for different user groups.

This command is used in conjunction with the **timeout** command.

Each user host's IP address has an authorization cache attached to it. If the user attempts to access a service that has been cached from the correct host, the firewall considers it preauthorized and immediately proxies the connection. This means that once you are authorized to access a website, for example, the authorization server is not contacted for each of the images as they are loaded (assuming they come from the same IP address). This significantly increases performance and reduces load on the authorization server.

The cache allows up to 16 address and service pairs for each user host.

The output from the **show uauth** command displays the username provided to the authorization server for authentication and authorization purposes, the IP address that the username is bound to, and whether the user is authenticated only, or has cached services.

Note Normally, when Xauth is enabled, an entry is added to the uauth table (as shown by the **show uauth/clear uauth** command) for the IP address assigned to the client. However, when using Xauth with the Easy VPN Remote feature in Network Extension Mode, the IPSec tunnel is created from network-to-network, so the users behind the firewall cannot be associated with a single IP address. For this reason, a uauth entry cannot be created upon completion of Xauth. If AAA authorization or accounting services are required, you can enable the AAA authentication proxy to authenticate users behind the firewall. For more information on AAA authentication proxies, please refer to the **aaa** commands.

Use the **timeout uauth** command to specify how long the cache should be kept after the user connections become idle. Use the **clear uauth** command to delete all authorization caches for all users, which will cause them to have to reauthenticate the next time they create a connection.

Examples

The following is sample output from the **show uauth** command when no users are authenticated and one user authentication is in progress:

```
pixfirewall(config)# show uauth
                        Current    Most Seen
Authenticated Users       0          0
Authen In Progress        0          1
```

The following is sample output from the **show uauth** command when three users are authenticated and authorized to use services through the PIX Firewall:

```
pixfirewall(config)# show uauth
user 'pat' from 209.165.201.2 authenticated
user 'robin' from 209.165.201.4 authorized to:
   port 192.168.67.34/telnet    192.168.67.11/http     192.168.67.33/tcp/8001
        192.168.67.56/tcp/25    192.168.67.42/ftp
user 'terry' from 209.165.201.7 authorized to:
   port 192.168.1.50/http     209.165.201.8/http
```

In this example, Pat has authenticated with the server but has not completed authorization. Robin has preauthorized connections to the Telnet, Web (HTTP), sendmail, FTP services, and to TCP port 8001 on 192.168.67.33.

Terry has been browsing the Web and is authorized for Web browsing to the two sites shown.

The next example causes Pat to reauthenticate:

```
clear uauth pat
```

Related Commands

aaa authorization	Enable or disable LOCAL or TACACS+ user authorization services.
timeout	Sets the maximum idle times.

show version

View the PIX Firewall operating information.

show version

Syntax Description

version	The PIX Firewall software version, hardware configuration, license key, and related uptime data.

Command Modes

Unprivileged mode.

Usage Guidelines

The **show version** command displays the PIX Firewall unit's software version, operating time since last reboot, processor type, Flash memory type, interface boards, serial number (BIOS ID), activation key value, license type (R or UR), and timestamp for when the configuration was last modified.

The serial number listed with the **show version** command in PIX Firewall software Version 5.3 and higher is for the Flash memory BIOS. This number is different from the serial number on the chassis. When you get a software upgrade, you will need the serial number that appears in the **show version** command, not the chassis number.

For PIX Firewall software Version 6.3(2) and higher, the **show version** command shows the maximum number of physical interfaces as well as the maximum number of logical interfaces for use with VLANs.

For PIX Firewall software Version 6.2 and higher, the **show version** command output appears as follows:

```
Running Activation Key: activation-key-four-tuple
```

to indicate the activation key that is currently running PIX Firewall image.

The amount of Flash memory is indicated at the end of the line showing the version of Flash installed on the PIX Firewall.

Throughput Limited indicates that the speed of the PIX Firewall interface is limited due to platform or version restrictions. ISAKMP peers Limited indicates that the number of IPSec peers is limited due to platform restrictions.

Note The uptime value indicates how long a failover set has been running. If one unit stops running, the uptime value will continue to increase as long as the other unit continues to operate.

Examples

The following is sample output from the **version** command:

```
pixfirewall(config)# show version

Cisco PIX Firewall Version 6.3(1)
Cisco PIX Device Manager Version 3.0(1)

Compiled on Wed 06-Nov-02 11:22 by root

pixfirewall up 4 days 22 hours

Hardware:   PIX-515E, 64 MB RAM, CPU Pentium 200 MHz
Flash i28F640J5 @ 0x300, 16MB
BIOS Flash AT29C257 @ 0xfffd8000, 32KB

0: ethernet0: address is 0003.e300.73fd, irq 10
1: ethernet1: address is 0003.e300.73fe, irq 7
2: ethernet2: address is 00d0.b7c8.139e, irq 9
Licensed Features:
Failover: Disabled
VPN-DES: Enabled
VPN-3DES-AES:       Disabled
Maximum Physical Interfaces: 6
Maximum Interfaces: 10
Cut-through Proxy:  Enabled
Guards:             Enabled
URL-filtering:      Enabled
Inside Hosts:       Unlimited
Throughput:         Unlimited
IKE peers:          Unlimited

This PIX has a Restricted (R) license.

Serial Number: 480430455 (0x1ca2c977)
```

```
Running Activation Key: 0xc2e94182 0xc21d8206 0x15353200 0x633f6734
Configuration last modified by enable_15 at 16:36:30.480 UTC Mon Nov 11 2002
```

Note The output of the **show version** command indicates whether the PIX Firewall has a Restricted (R) or Unrestricted (UR) license. A PIX Firewall with an R license cannot be used in a failover pair, and it has one half as much RAM as a PIX Firewall of the same platform with a UR license. Also, a PIX Firewall with an R license supports fewer physical interfaces and fewer logical interfaces (VLANs) than the same platform with a UR license. The number of interfaces allowed varies by platform.

show xlate/clear xlate

View or clear translation slot information.

clear xlate [**global** | **local** *ip1*[**-***ip2*] [**netmask** *mask*]] **lport** | **gport** *port*[**-***port*]] [**interface** *if1*[**,***if2*][**,***ifn*]] [**state static** [**,dump**] [**,portmap**] [**,norandomseq**] [**,identity**]]

show xlate [**detail**] [**global** | **local** *ip1* [**-***ip2*] [**netmask** *mask*]] **lport** | **gport** *port* [**-***port*]] [**interface** *if1* [**,***if2*] [**,***ifn*]] [**state static** [**,dump**] [**,portmap**] [**,norandomseq**] [**,identity**]] [**debug**] [**count**]

Syntax Description

detail	If specified, displays translation type and interface information.
[**global** | **local** *ip1* [**-***ip2*] [**netmask** *mask*]	Display active translations by global IP address or local IP address using the network mask to qualify the IP addresses.
interface *if1* [**,***if2*] [**,***ifn*]	Display active translations by interface.
lport | **gport** *port* [**-***port*]	Display active translations by local and global port specifications. See "Ports" in Chapter 2, "Using PIX Firewall Commands" for a list of valid port literal names.
state	Display active translations by state; **static** translation (**static**), **dump** (cleanup), PAT **global** (**portmap**), a **nat** or **static** translation with the **norandomseq** setting (**norandomseq**), or the use of the **nat 0**, identity feature (**identity**).
debug	Display translation type and interface information.
count	Display the number of active translations.

Command Modes

Privileged mode.

Usage Guidelines

The **clear xlate** command clears the contents of the translation slots. ("xlate" means translation slot.) The **show xlate** command displays the contents of only the translation slots.

Translation slots can persist after key changes have been made. Always use the **clear xlate** command after adding, changing, or removing the **aaa-server**, **access-list**, **alias**, **conduit**, **global**, **nat**, **route**, or **static** commands in your configuration.

Note When the **vpnclient** configuration is enabled and the inside host is sending out DNS requests, the **show xlate** command may list multiple xlates for a static translation.

The **show xlate detail** command displays the following information:

{**ICMP**|**TCP**|**UDP**} **PAT from** *interface*:*real-address*/*real-port* **to** *interface [acl-name]*:*mapped-address*/*mapped-port* **flags** *translation-flags*

NAT from *interface*:*real-address*/*real-port* **to** *interface [acl-name]*:*mapped-address*/*mapped-port* **flags** *translation-flags*

The translation flags are defined in Table 8-5.

Table 8-5 Translation Flags

Flag	Description
s	static translation slot
d	dump translation slot on next cleaning cycle
r	portmap translation (Port Address Translation)
n	no randomization of TCP sequence number
o	outside address translation
i	inside address translation
D	DNS A RR rewrite
I	identity translation from **nat 0**

Examples

The following is sample output from the **show xlate** command with three active Port Address Translations (PATs):

```
pixfirewall(config)# show xlate
3 in use, 3 most used
PAT Global 192.150.49.1(0) Local 10.1.1.15 ICMP id 340
PAT Global 192.150.49.1(1024) Local 10.1.1.15(1028)
PAT Global 192.150.49.1(1024) Local 10.1.1.15(516)
```

The following is sample output from the **show xlate detail** command with three active Port Address Translations (PATs):

```
pixfirewall(config)# show xlate detail
3 in use, 3 most used
Flags: D - DNS, d - dump, I - identity, i - inside, n - no random,
       o - outside, r - portmap, s - static
TCP PAT from inside:10.1.1.15/1026 to outside:192.150.49.1/1024 flags ri
UDP PAT from inside:10.1.1.15/1028 to outside:192.150.49.1/1024 flags ri
ICMP PAT from inside:10.1.1.15/21505 to outside:192.150.49.1/0 flags ri
```

The first entry is a TCP Port Address Translation for host-port (10.1.1.15, 1025) on the inside network to host-port (192.150.49.1, 1024) on the outside network. The flag "r" denotes the translation is a Port Address Translation. The "i" flags denotes that the translation applies to the inside address-port.

The second entry is a UDP Port Address Translation for host-port (10.1.1.15, 1028) on the inside network to host-port (192.150.49.1, 1024) on the outside network. The flag "r" denotes the translation is a Port Address Translation. The "i" flags denotes that the translation applies to the inside address-port.

The third entry is an ICMP Port Address Translation for host-ICMP-id (10.1.1.15, 21505) on the inside network to host-ICMP-id (192.150.49.1, 0) on the outside network. The flag "r" denotes the translation is a Port Address Translation. The "i" flags denotes that the translation applies to the inside address-ICMP-id.

The inside address fields appear as source addresses on packets traversing from the more secure interface to the less secure interface. Conversely, they appear as destination addresses on packets traversing from the less secure interface to the more secure interface.

The following is sample output from two static translations, the first with two associated connections (called “nconns”) and the second with four.

```
show xlate
Global 209.165.201.10 Local 209.165.201.10 static nconns 1 econns 0
Global 209.165.201.30 Local 209.165.201.30 static nconns 4 econns 0
```

The following is sample output from the **show xlate debug** command:

```
show xlate debug
1 in use, 1 most used
Flags: D - DNS, d - dump, I - identity, i - inside, n - no random,
       o - outside, r - portmap, s - static
NAT from inside:8.0.0.2 to outside:11.0.0.254 flags si idle 0:00:06 timeout 3:00:00
--------------------------------------------------------------------------------
```

Related Commands

show conn	Display all active connections.
show uauth/clear uauth	Display or delete all authorization caches for a user.
timeout	Sets the maximum idle times.

shun

The **shun** command enables a dynamic response to an attacking host by preventing new connections and disallowing packets from any existing connection.

[**no**] **shun** *src_ip* [*dst_ip sport dport* [*protocol*]]

clear shun [*statistics*]

show shun [*src_ip* | *statistics*]

Syntax Description

clear	Disable all shuns currently enabled and clears shun statistics. Specifying statistics only clears the counters for that interface.
dport	The destination port of the connection causing the shun.
dst_ip	The address of the of the target host.
no	Disable a shun based on *src_ip*, the actual address used by the PIX Firewall for shun lookups.

protocol	The optional IP protocol, such as UDP or TCP.
shun	Enable a blocking function (shun) based on *src_ip*.
sport	The source port of the connection causing the shun.
src_ip	The address of the attacking host.
statistics	Clear only interface counters.

Command Modes

Configuration mode.

Usage Guidelines

The **shun** command applies a blocking function to the interface receiving the attack. Packets containing the IP source address of the attacking host will be dropped and logged until the blocking function is removed manually or by the Cisco IDS master unit. No traffic from the IP source address will be allowed to traverse the PIX Firewall unit and any remaining connections will time out as part of the normal architecture. The blocking function of the **shun** command is applied whether or not a connection with the specified host address is currently active.

If the **shun** command is used only with the source IP address of the host, then the other defaults will be 0. No further traffic from the offending host will be allowed.

Because the **shun** command is used to block attacks dynamically, it is not displayed in your PIX Firewall configuration.

Examples

In the following example, the offending host (10.1.1.27) makes a connection with the victim (10.2.2.89) with TCP. The connection in the PIX Firewall connection table reads:

10.1.1.27, 555-> 10.2.2.89, 666 PROT TCP

If the **shun** command is applied in the following way:

```
shun 10.1.1.27 10.2.2.89 555 666 tcp
```

The preceding command would delete the connection from the PIX Firewall connection table, and it would also prevent packets from 10.1.1.27 from going through the PIX Firewall. The offending host can be inside or outside of the PIX Firewall.

The following is sample output of the **show shun** command with the **shun** command applied to the outside interface:

```
outside=ON, cnt=4,time=(0:04:13)
```

The first value indicates if the **shun** command is applied to the interface, the second value (**cnt**) indicates the number of packets that have been dropped since the **shun** command was applied. The third value (**time**) indicates the elapsed time since the **shun** command was applied to the interface.

sip ip-address-privacy

SIP address privacy provides the ability to hide phone IP addresses from one another. SIP fixup will retain outside IP addresses in the SIP header and SDP data of inbound packets. By default this command is turned off. When the command is turned on, SIP fixup will retain outside IP addresses in the SIP header and SDP data of inbound SIP packets.

sip ip-address privacy

[no] sip ip-address-privacy

Syntax Description

Field Name	Field Description
Via	The phone which originates the call puts in its IP address in the Via field.
From(if it contains an IP address)	If PAT is configured and this field does not contain a port, this field is not NATted (in the outbound direction).
Call-ID (if it contains an IP address)	If PAT is configured and this field does not contain a port, this field is not NATted.
o=	This contains the originator's IP address. This is a best effort in case of PAT. i.e, this field does not contain a port, so we do a 'best effort ' to PAT it by checking to see if it matches the connection address, and if it does, we use the m= port as the port to do the PAT. The SDP specification specifies the o= and m= as mandatory parameters in the SDP portion of the SIP packet. So, in a SIP packet conforming to the SDP specification, we will NAT/PAT the o= field with the port from the m= field (as described above).
·c=	This contains the connection IP address.
·m=	If PAT is configured, the PATted port should be retained.
	Record-route contains IP address.

Note By default, this feature is not turned on.

Command Modes

Global configuration

Usage Guidelines

The fixup can be enabled or disabled via the *[no]* **sip ip-address privacy** command.

Examples

INVITE sip:bob@Proxy SIP/2.0

Via: SIP/2.0/UDP A:5060 ==========> A':patport#

From: terry@A ==========> terry@A'

To: robin@Proxy

Call-ID:

Contact:terry@A ======================> terry@A'

SDP

o=A ==================================> A'

c=IN IP4 A ===========================> A'

m=port# ==============================> patport# (if applicable)

When the Proxy sends the INVITE to B:

INVITE sip:robin@Proxy SIP/2.0

Via: SIP/2.0/UDP A':5060 ===========>Has to remain as A':patport#

From: terry@A' ====================>Has to remain as A'

To:robin@Proxy

Call-ID:

Contact:terry@A' ==================>Has to remain as A'

SDP

o=A' ============================>Has to remain as A'

c=IN IP4 A' ======================>Has to remain as A'

m=patport#

Note When this feature is turned on outside NAT will not work. When a packet from the lower security level (eg., outside) comes to the higher security level (eg., inside), since we retain the NATted IP addresses in it and don't send the packet through the NAT engine, outside NAT will not be performed for the inbound SIP packets.

- ·When this feature is off, regular SIP Fixup will work as it does under PIX 6.3.3
- ·When this feature is turned on with sip ip-address privacy, all messages/responses are inspected and NATted IP addresses are retained for all relevant fields.
- · RTP traffic between phones on the same interface must go through the PIX Firewall. Thus, necessary pinholes for RTP traffic must be opened on the PIX.

Related Commands **show running-config** can be used to see if the **sip ip-address privacy** command is turned on. Debug messages are available when outside IP addresses are retained in a system message when this feature is enabled.

snmp deny version

snmp deny version filters out traffic based on the protocol version field in SNMP packets with the variable <version-string>. To disable, use the **no** form of this command.

[no] snmp deny version [1 | 2 | 2c | 3]

Syntax Description

1	Specifies SNMP Version 1.
2	Specifies SNMP Version 2.
2c	Specifies SNMP Version 2c.
3	Specifies SNMP Version 3.

Defaults

No default behavior or values.

Command Modes

Global configuration

Usage Guidelines

The fixup can be enabled or disable via the fixup cmd:

[no] fixup protocol snmp 161-162

Note Existing connections will retain present fixup configurations fromtheir initial creation.

So, if you toggle the configuration, you need to either:

- Wait for the connections to time out
- Manually clear the connections

Use **clear xlate** or **clear local** to clear connections for the fixup configuration to take effect.

fixup protocol sqlnet

PIX Firewall uses port 1521 for SQL*Net. This is the default port used by Oracle for SQL*Net; however, this value does not agree with IANA port assignments.

Examples

The following example filters out SNMP Version 2c traffic:

```
pix# snmp deny version 2c
```

Related Commands

fixup protocol snmp

snmp-server

Provide PIX Firewall event information through SNMP.

[**no**] **snmp-server community** *key*

[**no**] **snmp-server** {**contact** | **location**} *text*

[**no**] **snmp-server host** [*if_name*] *ip_addr* [**trap** | **poll**]

[**no**] **snmp-server enable traps**

clear snmp-server

show snmp-server

Syntax Description

community *key*	Enter the password key value in use at the SNMP management station. The SNMP community string is a shared secret among the SNMP management station and the network nodes being managed. PIX Firewall uses the key to determine if the incoming SNMP request is valid. For example, you could designate a site with a community string and then configure the routers, firewall, and the management station with this same string. The PIX Firewall then honors SNMP requests using this string and does not respond to requests with an invalid community string. The *key* is a case-sensitive value up to 32 characters in length. Spaces are not permitted. The default is **public** if *key* is not set. Consequently, it is important to specify a (new) value for *key* for security reasons.
contact *text*	Supply your name or that of the PIX Firewall system administrator. The text is case-sensitive and can be up to 127 characters. Spaces are accepted, but multiple spaces are shortened to a single space.
enable traps	Enable or disable sending log messages as SNMP trap notifications.
host	Specify an IP address of the SNMP management station to which traps should be sent and/or from which the SNMP requests come. You can specify up to 32 SNMP management stations.
if_name	The interface name where the SNMP management station resides.
ip_addr	The IP address of a host to which SNMP traps should be sent and/or from which the SNMP requests come.
location *text*	Specify your PIX Firewall location. The text is case-sensitive and can be up to 127 characters. Spaces are accepted, but multiple spaces are shortened to a single space.
trap | **poll**	Specify whether traps, polls, or both are acted upon. Use with these parameters: • **trap**—Only traps will be sent. This host will not be allowed to poll. • **poll**—Traps will not be sent. This host will be allowed to poll. The default allows both traps and polls to be acted upon.

Command Modes

Configuration mode.

Usage Guidelines

Use the **snmp-server** command to identify site, management station, community string, and user information.

Note In the **snmp-server community** *key* command, the default value for *key* is **public**. Consequently, it is important to specify a (new) value for *key* for security reasons.

The **clear snmp-server** and **no snmp-server** commands disable the SNMP commands in the configuration as follows:

```
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
```

In understanding SNMP use, the PIX Firewall is considered the SNMP agent or SNMP server. The management station is the system running the SNMP program that receives and processes the SNMP information that the PIX Firewall sends.

An SNMP object ID (OID) for PIX Firewall displays in SNMP event traps sent from the PIX Firewall. The OIDs for the PIX Firewall platforms are listed in Table 8-6.

Table 8-6 System OID in PIX Firewall Platforms

PIX Firewall Platform	System OID
PIX 501	.1.3.6.1.4.1.9.1.417
PIX 506	.1.3.6.1.4.1.9.1.389
PIX 506E	.1.3.6.1.4.1.9.1.450
PIX 515	.1.3.6.1.4.1.9.1.390
PIX 515E	.1.3.6.1.4.1.9.1.451
PIX 520	.1.3.6.1.4.1.9.1.391
PIX 525	.1.3.6.1.4.1.9.1.392
PIX 535	.1.3.6.1.4.1.9.1.393
Others	.1.3.6.1.4.1.9.1.227

Use the **trap** and **poll** command options to configure hosts to participate only in specific SNMP activities. Poll responses and traps are sent only to the configured entities. Hosts configured with the **trap** command option will have traps sent to them, but will not be allowed to poll. Hosts configured with the **poll** command option will be allowed to poll, but will not have traps sent to them. Refer to the *Cisco PIX Firewall and VPN Configuration Guide* for more information on how to access and monitor the PIX Firewall using SNMP traps.

Accessibility to PIX Firewall Management Information Bases (MIBs) is based on configuration, MIB support, and authentication based on the community string. Unsuccessful polling attempts, except for failed community string authentication, are not logged or otherwise indicated. Community authentication failures result in a trap where applicable.

MIB Support

You can browse the System and Interface groups of MIB-II. All SNMP values in the PIX Firewall are read only (RO). The PIX Firewall does not support browsing of the Cisco syslog MIB.

Browsing a MIB is different from sending traps. Browsing means doing an **snmpget** or **snmpwalk** of the MIB tree from the management station to determine values. Traps are different; they are unsolicited "comments" from the managed device to the management station for certain events, such as link up, link down, syslog event generated, and so on.

The Cisco Firewall MIB, Cisco Memory Pool MIB, Cisco Process MIB provide the following PIX Firewall information through SNMP:

- Buffer usage from the **show block** command
- Connection count from the **show conn** command
- CPU usage through the **show cpu usage** command
- Failover status
- Memory usage from the **show memory** command

Receiving SNMP Requests from an SNMP Management Station

To receive SNMP requests from a management station, perform the following steps:

Step 1 Identify the management station with an **snmp-server host** command statement.

Step 2 Specify **snmp-server** command options for the **location**, **contact**, and **community**.

Step 3 Start the SNMP software on the management station and begin issuing SNMP requests to the PIX Firewall.

Defaults

If you do not specify an option, the **snmp-server host** command behaves as in previous versions. The polling is permitted from all configured hosts on the affected interface. Traps are sent to all configured hosts on the affected interface.

Examples

The following example shows commands you would enter to start receiving SNMP requests from a management station:

```
snmp-server community wallawallabingbang
snmp-server location Building 42, Sector 54
snmp-server contact Sherlock Holmes
snmp-server host perimeter 10.1.2.42
```

The next example is sample output from the **show snmp-server** command:

```
show snmp
snmp-server host perimeter 10.1.2.42
snmp-server location Building 42, Sector 54
snmp-server contact Sherlock Holmes
snmp-server community wallawallabingbang
```

ssh

Specify a host for PIX Firewall console access through Secure Shell (SSH).

[**no**] **ssh** *ip_address* [*netmask*] [*interface_name*]

ssh timeout *mm*

ssh disconnect *session_id*

clear ssh

show ssh [**sessions** [*ip_address*]]

show ssh timeout

Syntax Description

interface_name	PIX Firewall interface name on which the host or network initiating the SSH connection resides.
ip_address	IP address of the host or network authorized to initiate an SSH connection to the PIX Firewall.
mm	The duration in minutes that a session can be idle before being disconnected. The default duration is 5 minutes. The allowable range is from 1 to 60 minutes.
netmask	Network mask for *ip_address*. If you do not specify a *netmask*, the default is 255.255.255.255 regardless of the class of *ip_address*.
session_id	SSH session ID number, viewable with the **show ssh sessions** command.

Command Modes

Configuration mode.

Usage Guidelines

The **ssh** *ip_address* command specifies the host or network authorized to initiate an SSH connection to the PIX Firewall. The **ssh timeout** command lets you specify the duration in minutes that a session can be idle before being disconnected. The default duration is 5 minutes. Use the **show ssh sessions** command to list all active SSH sessions on the PIX Firewall. The **ssh disconnect** command lets you disconnect a specific session you observed from the **show ssh sessions** command. Use the **clear ssh** command to remove all **ssh** command statements from the configuration. Use the **no ssh** command to remove selected **ssh** command statements from the configuration.

Note You must generate an RSA key-pair for the PIX Firewall before clients can connect to the PIX Firewall console. After generating the RSA key-pair, save the key-pair using the **ca save all** command. To use SSH, your PIX Firewall must have a DES or 3DES activation key.

To gain access to the PIX Firewall console via SSH, at the SSH client, enter the username as **pix** and enter the Telnet password. You can set the Telnet password with the **passwd** command; the default Telnet password is **cisco**. To authenticate using the AAA server instead, configure the **aaa authenticate ssh console** command.

SSH permits up to 100 characters in a username and up to 50 characters in a password.

When starting an SSH session, a dot (.) displays on the PIX Firewall console before the SSH user authentication prompt appears.

The dot appears as follows:

```
pixfirewall(config)# .
pixfirewall(config)# .
```

The display of the dot does not affect the functionality of SSH. The dot appears on at the console when generating a server key or decrypting a message using private keys during SSH key exchange, before user authentication occurs. These tasks can take up to two minutes or longer. The dot is a progress indicator that verifies that the PIX Firewall is busy and has not hung.

show ssh sessions Command

The **show ssh sessions** command provides the following display:

```
Session ID      Client IP       Version Encryption      State   Username
    0           172.16.25.15    1.5     3DES            4       -
    1           172.16.38.112   1.5     DES             6       pix
    2           172.16.25.11    1.5     3DES            4       -
```

The Session ID is a unique number that identifies an SSH session. The Client IP is the IP address of the system running an SSH client. The Version lists the protocol version number that the SSH client supports. The Encryption column lists the type of encryption the SSH client is using. The State column lists the progress the client is making as it interacts with the PIX Firewall. The Username column lists the login username that has been authenticated for the session. The "pix" username appears when non-AAA authentication is used.

The following table lists the SSH states that appear in the State column:

Number	SSH State
0	SSH_CLOSED
1	SSH_OPEN
2	SSH_VERSION_OK
3	SSH_SESSION_KEY_RECEIVED
4	SSH_KEYS_EXCHANGED
5	SSH_AUTHENTICATED
6	SSH_SESSION_OPEN
7	SSH_TERMINATE
8	SSH_SESSION_DISCONNECTING
9	SSH_SESSION_DISCONNECTED
10	SSH_SESSION_CLOSED

SSH Syslog Messages

Syslog messages 315001, 315002, 315003, 315004, 315005, and 315011 were added for SSH. Refer to *Cisco PIX Firewall System Log Messages* for more information.

Obtaining an SSH Client

The following sites let you download an SSH v1.*x* client. Because SSH Version 1.*x* and 2 are entirely different protocols and are not compatible, be sure you download a client that supports SSH v1.*x*.

- Windows 3.1, Windows CE, Windows 95, and Windows NT 4.0—download the free Tera Term Pro SSH v1.x client from the following website:

  http://hp.vector.co.jp/authors/VA002416/teraterm.html

  The TTSSH security enhancement for Tera Term Pro is available at the following website:

  http://www.zip.com.au/~roca/ttssh.html

  Note You must download TTSSH to use Tera Term Pro with SSH. TTSSH provides a Zip file you copy to your system. Extract the zipped files into the same folder that you installed Tera Term Pro. For a Windows 95 system, by default, this would be the C:\Program Files\Ttempro folder.

- Linux, Solaris, OpenBSD, AIX, IRIX, HP/UX, FreeBSD, and NetBSD—download the SSH v1.x client from the following website:

  http://www.openssh.com

- Macintosh (international users only)—download the Nifty Telnet 1.1 SSH client from the following website:

  http://www.lysator.liu.se/~jonasw/freeware/niftyssh/

Changed aaa Command for SSH

The **aaa** command adds the **ssh** option for use with SSH:

aaa authentication [**serial** | **enable** | **telnet** | **ssh**] **console** *group_tag*

The new **ssh** option specifies the group of AAA servers to be used for SSH user authentication. The authentication protocol and AAA server IP addresses are defined with the **aaa-server** command statement.

Similar to the Telnet model, if the **aaa authentication ssh console** *group_tag* command statement is not defined, you can gain access to the PIX Firewall console with the username **pix** and with the PIX Firewall Telnet password (set with the **passwd** command). If the **aaa** command is defined, but the SSH authentication request times out, this implies that the AAA server may be down or not available. You can gain access to the PIX Firewall using the username **pix** and the enable password (set with the **enable password** command). By default, the Telnet password is **cisco** and the enable password is not set. If the enable password is empty (null), even if you enter the password correctly, you are not granted access to the SSH session.

The user authentication attempt limit is set to 3. Note that the Linux version of the SSH Version 1 client available from http://www.openssh.com only allows one user authentication attempt.

Examples

Create an RSA key-pair with a modulus size of 1024 bits (recommended for use with Cisco IOS software):

```
hostname cisco-pix
domain-name example.com
ca generate rsa key 1024
show ca mypubkey rsa
ca save all
```

These command statements set the host name and domain name for the PIX Firewall, generate the RSA key-pair, display the RSA key-pair, and save the RSA key-pair to Flash memory.

Start an SSH session so clients on the outside interface can access the PIX Firewall console remotely over a secure shell:

```
ssh 10.1.1.1 255.255.255.255 outside
ssh timeout 60
```

Configure the PIX Firewall to perform user authentication using AAA servers. The protocol is the protocol used by the AAA-server to perform the authentication. The following example uses the TACACS+ authentication protocol.

```
aaa-server ssh123 (inside) host 10.1.1.200 mysecure
aaa-server ssh123 protocol tacacs+
aaa authenticate ssh console ssh123
```

Related Commands

- **aaa accounting**
- ca
- domain-name
- **hostname**
- **password**

static

Configure a one-to-one address translation rule by mapping a local IP address to a global IP address, or a local port to a global port.

[**no**] **static** [(*local_ifc*,*global_ifc*)] {*global_ip* | **interface**} {*local_ip* [**netmask** *mask*] | **access-list** *acl_name*} [**dns**] [**norandomseq**] [*max_conns* [*emb_limit*]]

[**no**] **static** [(*local_ifc*,*global_ifc*)] {**tcp** | **udp**} {*global_ip* | **interface**} *global_port* {*local_ip local_port* **[netmask** *mask***]** | **access-list** *acl_name*} [**dns**] [**norandomseq**] [*max_conns* [*emb_limit*]]

show static

Syntax Description

access-list	Lets you identify local traffic for network address translation (NAT) by specifying the local and destination addresses (or ports). This feature is known as policy NAT. The subnet mask used in the access list is also used for the *global_ip*. You can only include **permit** statements in the access list.
acl_name	Specifies the access list name.
dns	Rewrites the local address in DNS replies to the global address.

emb_limit	Specifies the maximum number of embryonic connections per host. An embryonic connection is a connection request that has not finished the necessary handshake between source and destination. Set a small value for slower systems, and a higher value for faster systems. The default is 0, which means unlimited embryonic connections. The embryonic connection limit lets you prevent a type of attack where processes are started without being completed. When the embryonic limit is surpassed, the TCP intercept feature intercepts TCP synchronization (SYN) packets from clients to servers on a higher security level. 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 combines the two half-connections together transparently. Thus, connection attempts from unreachable hosts never reach the server. The PIX firewall accomplishes TCP intercept functionality using SYN cookies. **Note** This option does not apply to outside NAT. The TCP intercept feature applies only to hosts or servers on a higher security level. If you set the embryonic limit for outside NAT, the embryonic limit is ignored.
global_ifc	Specifies the interface where you want to use the global address. For example, if you want to translate an inside address when it exits the outside interface, then the outside interface is the global interface. If this interface is a higher security level than the local interface, then this translation is known as outside NAT. Some options do not apply to outside NAT.
global_ip	Specifies the global IP address(es) to which you want to translate the local address(es). You can map a single global address to a single local address, or map a range of global addresses to a range of local addresses. This address cannot be used as a dynamic Port Address Translation (PAT) IP address in the **global** command unless you use static PAT, in which case the two addresses can be the same.
global_port	Specifies the global TCP or UDP port. You can specify ports by either a literal name or a number in the range of 0 to 65535. You can view valid port numbers online at the following website: http://www.iana.org/assignments/port-numbers See "Ports" in "Using PIX Firewall Commands" for a list of valid port literal names in port ranges; for example, **ftp** or **h323**. You can also specify numbers.
interface	Specifies the interface IP address for the global address. Use this keyword if you want to use the interface address, but the address is dynamically assigned using DHCP.
local_ifc	Specifies the interface that is connected to the local address. For example, if you want to translate an inside address when it exits the outside interface, then the inside interface is the local interface. If this interface is a lower security level than the global interface, then this translation is known as outside NAT. Some options do not apply to outside NAT (such as **norandomseq** and *emb_limit*).
local_ip	Specifies the addresses to translate. You can map a single local address to a single global address or map a range of local addresses to a range of global addresses.

local_port	Specifies the local TCP or UDP port. You can specify ports by either a literal name or a number in the range of 0 to 65535. You can view valid port numbers online at the following website: http://www.iana.org/assignments/port-numbers See "Ports" in "Using PIX Firewall Commands" for a list of valid port literal names in port ranges; for example, **ftp** or **h323**. You can also specify numbers.
mask	Specifies the network mask used for both *global_ip* and *local_ip*. For single hosts, use 255.255.255.255. If you use the **access-list** option instead of the *local_ip*, then the subnet mask used in the access list is also used for the *global_ip*.
max_conns	Specifies the maximum number of simultaneous TCP and UDP connections for the entire subnet. The default is 0, which means unlimited connections. (Idle connections are closed after the idle timeout specified by the **timeout conn** command.) **Note** This option does not apply to outside NAT. The firewall only tracks connections from a higher security interface to a lower security interface. If you set *max_conns* for outside NAT, the *max_conns* option is ignored.
netmask	Specifies the keyword required before specifying the network mask. If you do not enter a mask, then the default mask for the IP address class is used.
norandomseq	Disables TCP Initial Sequence Number (ISN) randomization protection. Only use this option if another inline firewall is also randomizing sequence numbers and the result is scrambling the data. Without this protection, inside hosts with weak self-ISN protection become more vulnerable to TCP connection hijacking. **Note** This option does not apply to outside NAT. The firewall only randomizes the ISN that is generated by the host/server on the higher security interface. If you set **norandomseq** for outside NAT, the **norandomseq** option is ignored.
tcp	Specifies a TCP port.
udp	Specifies a UDP port.

Command Modes

Configuration mode.

Usage Guidelines

The **static** command creates a one-to-one address translation rule (called a static translation slot or "xlate"). Each local address is translated to a fixed global address. With dynamic NAT and PAT, each host uses a different address or port for each consecutive connection. Because the global address is the same for each consecutive connection, and a persistent translation rule exists, the **static** command allows hosts on the global network to initiate traffic to a local host (if the access list allows it).

Static Port Address Translation (PAT) is the same as static NAT, except it allows you to specify the protocol (TCP or UDP) and port for the local and global addresses.

After changing or removing a **static** command statement, use the **clear xlate** command to clear the translations.

Unless you use static PAT, you cannot create multiple **static** commands with the same global IP addresses.

Static Port Address Translation (Static PAT)

This feature allows you to identify the same global address across many different static statements, so long as the port is different for each statement (you cannot use the same global address for multiple static *NAT* statements). For example, if you want to provide a single address for global users to access FTP, HTTP, and SMTP, but these are all actually different servers on the local network, you can specify static statements for:

- local_ip_A/global_ip_A/FTP
- local_ip_B/global_ip_A/HTTP
- local_ip_C/global_ip_A/SMTP

You can also use this feature to translate a well-known port to a lesser-known port or vice versa. For example, if your inside web servers use port 8080, you can allow outside users to connect to port 80, and then translate them to the correct port. Similarly, if you want to provide extra security, you can tell your web users to connect to lesser-known port 6785, and then translate them to port 80 on the local network.

Note PIX Firewall Version 6.2 introduced support for PAT and static PAT of H.323 application traffic; PAT is not supported for H.323 in earlier versions.

Static PAT supports all applications that are supported by dynamic PAT, including the same application constraints. The Telnet port 23 and PFM port 1467 of the PIX Firewall interface cannot be used for Static PAT because the PIX Firewall requires that traffic to these ports be protected by IPSec.

static access-list (Policy NAT)

When you use an access list with the **static** command, then you enable policy NAT.

Policy NAT lets you identify local traffic for address translation by specifying the source and destination addresses (or ports) in an access list. Regular NAT uses source addresses/ports only, whereas policy NAT uses both source and destination addresses/ports.

With policy NAT, you can create multiple **static** statements that identify the same local address as long as the source/port and destination/port combination is unique for each statement. You can then match different global addresses to each source/port and destination/port pair.

While static PAT already allowed you to identify the local and global ports, policy NAT enhances this feature (as well as static NAT) by allowing you to identify the destination address for the local traffic.

Identity NAT

If you want to bypass NAT and allow the local address to appear unchanged on the global network, you can enter the same address for the local and global addresses:

static (*local_ifc*, *global_ifc*) *local_ip local_ip ...*

You can use policy NAT with identity NAT to bypass NAT only for traffic going to a particular destination.

Permitting Inbound Traffic with Access Lists

In addition to using the **static** command, you must also use an **access-list** command to allow outside traffic to access inside hosts or servers.

For example, the host you want to make accessible on the dmz2 network is 192.168.1.1. The static command maps this address to 10.1.1.1:

```
static (dmz2,dmz1) 10.1.1.1 192.168.1.1 netmask 255.255.255.255
```

The **access-list** and **access-group** commands allow traffic from the dmz1 network to access this host on the dmz2 network. Note that the host that dmz1 users want to access is the translated global address 10.1.1.1.

```
access-list acl_dmz1 permit tcp 10.1.1.0 255.255.255.0 host 10.1.1.1
access-group acl_dmz1 in interface dmz1
```

Note Always make **access-list** command statements as specific as possible. Using the **any** option to allow any host access should be used with caution for access lists used with statics.

Order of NAT Commands Used to Match Local Addresses

The firewall matches local traffic to NAT commands in the following order:

1. **nat 0 access-list** (NAT exemption)—In order, until the first match. For example, you could have overlapping local/destination addresses in multiple **nat** commands, but only the first command is matched.
2. **static** (static NAT)—In order, until the first match. Because you cannot use the same local address in static NAT or static PAT commands, the order of **static** commands does not matter. Similarly, for static policy NAT, you cannot use the same local/destination address and port across multiple statements.
3. **static** {**tcp** | **udp**} (static PAT)—In order, until the first match. Because you cannot use the same local address in static NAT or static PAT commands, the order of **static** commands does not matter. Similarly, for static policy NAT, you cannot use the same local/destination address and port across multiple statements.
4. **nat** *nat_id* **access-list** (policy NAT)—In order, until the first match. For example, you could have overlapping local/destination ports and addresses in multiple **nat** commands, but only the first command is matched.
5. **nat** (regular NAT)—Best match. The order of the NAT commands does not matter. The **nat** statement that best matches the local traffic is used. For example, you can create a general statement to translate all addresses (0.0.0.0) on an interface. If you also create a statement to translate only 10.1.1.1, when 10.1.1.1 makes a connection, the specific statement for 10.1.1.1 is used because it matches the local traffic best.

Failover and the static command

The **static** command, without a port specified, translates all traffic received on the interface, including failover messages sent by a standby failover unit. In this case, the standby failover unit sends messages to the active unit, but they bypass the active unit, so the standby failover unit receives no replies from the active unit and it assumes that the interface is down and becomes the active unit. When you specify the port number, only traffic to that port is translated, and this situation is avoided. (Because failover uses a unique port number, port 105, it is not translated when other specific ports are.)

statics and VoIP

In networks with VoIP traffic, pay close attention to any static translations in your configuration. VoIP calls can fail to pass through the firewall if, after configuring a static translation for a network, the third party endpoint has a global IP address that matches the static translation. For example, if the IP addresses are as follows:

```
inside IP phone: 10.132.60.231
outside IP phone 10.130.60.215
outside CM: 10.130.60.111
```

and the following command is used:

```
static (inside,outside) 10.130.60.0 10.132.60.0 netmask 255.255.255.0
```

Then, when the firewall receives a message from the outside CM to the inside phone, the firewall sees the outside phone's IP address as a global IP address of an inside phone and translates it (so the call does not connect).

TCP Intercept

Prior to Version 5.3, the PIX Firewall offered no mechanism to protect systems reachable via a static and TCP conduit from TCP SYN attacks. Previously, if an embryonic connection limit was configured in a **static** command statement, the firewall simply dropped new connection attempts once the embryonic threshold was reached. Given this, a modest attack could stop web traffic. For **static** command statements without an embryonic connection limit, the firewall passes all traffic. If the affected system does not have TCP SYN attack protection, and most operating systems do not offer sufficient protection, then the affected system's embryonic connection table overloads and all traffic stops.

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

TCP intercept requires no change to the PIX Firewall command set. Note only that the embryonic connection limit on the **static** command now has a new behavior.

Note The TCP intercept feature applies only to hosts or servers on a higher security level. If you set the embryonic limit for outside NAT, the embryonic limit is ignored.

Deny Xlate for Network or Broadcast Address for Inbound Traffic

For all inbound traffic, the firewall denies translations for destination IP addresses identified as network address or broadcast addresses. The firewall utilizes the global IP and mask from a **static** command statement to differentiate regular IP addresses from network or broadcast addresses. If a global IP address is a valid network address with a matching network mask, then the firewall disallows the translation for network or broadcast IP addresses with inbound packet.

Interfaces on Which to Use Static NAT or Dynamic NAT

The rules for which command to use with an interface is summarized in Table 8-7. Table 8-7 assumes that the security levels are 40 for dmz1 and 60 for dmz2.

Table 8-7 Interface Access Commands by Interface

From This Interface	To This Interface	Use This Command
inside	outside	**nat**
inside	dmz1	**nat**
inside	dmz2	**nat**
dmz1	outside	**nat**
dmz1	dmz2	**static**
dmz1	inside	**static**
dmz2	outside	**nat**
dmz2	dmz1	**nat**
dmz2	inside	**static**
outside	dmz1	**static**
outside	dmz2	**static**
outside	inside	**static**

Examples

Basic Static NAT Examples

The following example permits a finite number of users to call in through H.323 using an Intel Internet Phone, CU-SeeMe, CU-SeeMe Pro, MeetingPoint, or MS NetMeeting. The **static** command maps addresses 209.165.201.1 through 209.165.201.30 to local addresses 10.1.1.1 through 10.1.1.30 (209.165.201.2 maps to 10.1.1.2, 209.165.201.10 maps to 10.1.1.10, and so on). The accompanying **access-list** and **access-group** commands allow traffic from a lower security interface to a higher security interface.

```
static (inside,outside) 209.165.201.0 10.1.1.0 netmask 255.255.255.224
access-list acl_out permit tcp any 209.165.201.0 255.255.255.224 eq h323
access-group acl_out in interface outside
```

The following example shows the commands used to disable Mail Guard:

```
static (dmz1,outside) 209.165.201.1 10.1.1.1 netmask 255.255.255.255
access-list acl_out permit tcp any host 209.165.201.1 eq smtp
access-group acl_out in interface outside
no fixup protocol smtp 25
```

In this example, the **static** command sets up a global address to permit outside hosts access to the 10.1.1.1 mail server host on the dmz1 interface. (The MX record for DNS needs to point to the 209.165.201.1 address so that mail is sent to this address.) The **access-list** command lets any outside users access the global address through the SMTP port (25). The **no fixup protocol** command disables Mail Guard.

Static PAT Examples

To redirect Telnet traffic from the PIX Firewall outside interface to the inside host at 10.1.1.15, enter:

```
static (inside,outside) tcp interface telnet 10.1.1.15 telnet netmask 255.255.255.255
```

To redirect FTP traffic from the PIX Firewall outside interface to the inside host at 10.1.1.30, enter:

```
static (inside,outside) tcp interface ftp 10.1.1.30 ftp netmask 255.255.255.255
```

To redirect DNS traffic from the PIX Firewall outside interface to the inside host at 10.1.1.30, enter:

```
static (inside,outside) udp interface domain 10.1.1.30 domain netmask 255.255.255.255
```

To allow the local Telnet server to initiate connections other than Telnet, you need to provide additional translation. For example, to translate all other types of traffic to the same address used in the static translation for Telnet (the interface address, for example), enter the following commands:

```
static (inside,outside) tcp 10.1.2.14 telnet 10.1.1.15 telnet netmask 255.255.255.255
nat (inside) 1 10.1.1.15 255.255.255.255
global (outside) 1 10.1.2.14 netmask 255.255.255.255
```

The **static** command provides the translation for Telnet. The **nat** and **global** commands provide PAT for all other outbound connections from the server.

If you have a separate translation for all inside traffic that uses a different global address, you can still configure the Telnet server to use the same address as the static statement by creating a more exclusive **nat** statement just for that server. Because **nat** statements are read for the best match, more exclusive **nat** statements are matched before general statements.

```
static (inside,outside) tcp 10.1.2.14 telnet 10.1.1.15 telnet netmask 255.255.255.255
nat (inside) 1 10.1.1.15 255.255.255.255
global (outside) 1 10.1.2.14 netmask 255.255.255.255
nat (inside) 2 0.0.0.0 0.0.0.0
global (outside) 2 10.1.2.78 netmask 255.255.255.255
```

To translate a well-known port (80) to another port (8080), enter:

```
static (inside,outside) tcp 10.1.2.45 80 10.1.1.16 8080 netmask 255.255.255.255
```

Policy NAT Examples

The following example shows a Policy NAT configuration. In this example, traffic destined for the 172.16.1.0/24 from host 10.1.1.10 is translated as 192.150.49.10, and traffic destined for the 172.16.2.0/24 from host 10.1.1.10 is translated as 192.150.49.20:

```
access-list network-1 permit ip host 10.1.1.10 172.16.1.0 255.255.255.0
access-list network-2 permit ip host 10.1.1.10 172.16.2.0 255.255.255.0
static (inside,outside) 192.150.49.10 access-list network-1
static (inside,outside) 192.150.49.20 access-list network-2
```

If you want to use identity NAT from traffic going from 10.1.1.1 to 10.2.2.3, but you want to translate 10.1.1.1 to 10.4.5.6 when going to 10.3.1.0/24, you could enter:

```
access-list IDENTITY permit ip host 10.1.1.1 host 10.2.2.3
access-list TRANSLATE permit ip host 10.1.1.1 10.3.1.0 255.255.255.0
static (inside,outside) 10.1.1.1 access-list IDENTITY
static (inside,outside) 10.4.5.6 access-list TRANSLATE
```

Identity NAT Examples

For example, a web server on the **dmz**, 209.165.201.5 needs to be accessible by users on the outside. The **static** and **access-list** command statements are as follows:

```
static (dmz,outside) 209.165.201.5 209.165.201.5 netmask 255.255.255.255
access-list acl_out permit tcp any host 209.165.201.5 eq www
access-group acl_out in interface outside
```

The **static** command presents the 209.165.201.5 address on the outside interface. The DNS server on the outside would map this IP address to the domain of the company; for example, example.com. Users accessing example.com are permitted to access the web server via port 80 by the **access-list** command.

Another example of identity NAT statics is when users on dmz1 need to access a web server on dmz2. The network uses a Class C address and the .240 subnet. Addresses 209.165.201.1 to 209.165.201.14 are on dmz1, and addresses 209.165.201.17 to 209.165.201.30 are on dmz2. The web server is at 209.165.201.25. The **static** and **access-list** command statements are as follows:

```
static (dmz2,dmz1) 209.165.201.25 209.165.201.25 netmask 255.255.255.255
access-list acl_dmz1 permit tcp any host 209.165.201.25 eq www
access-group acl_dmz1 in interface dmz1
```

The **static** command statement opens access to the web server at 209.165.201.25. The **access-list** command statement permits access to the web server only on port 80 (**www**).

Related Commands

- **access-list**
- **show xlate/clear xlate**

syslog

Enable syslog message facility. Obsolete command replaced by the **logging** command.

See the **logging** command for more information. The **syslog** command is available for backward compatibility.

sysopt

Change PIX Firewall system options.

[no] sysopt connection {permit-pptp | permit-l2tp | permit-ipsec}

[no] sysopt connection tcpmss [minimum] *bytes*

[no] sysopt connection timewait

[no] sysopt ipsec pl-compatible

[no] sysopt nodnsalias {inbound | outbound}

[no] sysopt noproxyarp *if_name*

[no] sysopt radius ignore-secret

[no] sysopt uauth allow-http-cache

clear sysopt

show sysopt

Syntax Description

connection permit-ipsec	Implicitly permit any packet that came from an IPSec tunnel and bypass the checking of an associated **access-list, conduit,** or **access-group** command statement for IPSec connections.
connection permit-l2tp	Implicitly permit any packet that came from an L2TP/IPSec tunnel and bypass the checking of an associated **access-list, conduit,** or **access-group** command statement for L2TP/IPSec connections.
connection permit-pptp	Allow PPTP traffic to bypass **conduit** or **access-list** command statement checking.
connection tcpmss [minimum] *bytes*	Overrides the maximum TCP segment size to be no greater than *bytes*. The **minimum** keyword overrides the maximum segment size to be no less than *bytes*. The minimum value is 48 bytes. The default value is 1380 bytes.
connection timewait	Force each TCP connection to linger in a shortened TIME_WAIT state of at least 15 seconds after the final normal TCP close-down sequence.
ipsec pl-compatible	Enable IPSec packets to bypass the PIX Firewall unit's NAT and ASA features and allows incoming IPSec packets to terminate on the inside interface.
nodnsalias inbound	Disable inbound embedded DNS A record fixups according to aliases that apply to the A record address.
nodnsalias outbound	Disable outbound DNS A record replies.
noproxyarp *if_name*	Disable proxy-ARPs on a PIX Firewall interface.
radius ignore-secret	Ignore authenticator key to avoid retransmit caveat.
uauth allow-http-cache	Allows the web browser to supply a username and password from its cache for AAA authentication.

Command Modes

Configuration mode.

Usage Guidelines

The **sysopt** commands let you tune various PIX Firewall security and configuration features. In addition, you can use this command to disable the PIX Firewall IP Frag Guard feature.

There is no need to enter the **sysopt connection permit-12tp** command if the **sysopt connection permit-ipsec** command is present.

sysopt connection permit-ipsec

Use the **sysopt connection permit-ipsec** command in IPSec configurations to permit IPSec traffic to pass through the PIX Firewall without a check of **conduit** or **access-list** command statements.

An **access-list** or **conduit** command statement must be available for inbound sessions.

By default, any inbound session must be explicitly permitted by a **conduit** or **access-list** command statement. With IPSec protected traffic, the secondary access list check could be redundant. To enable IPSec authenticated/cipher inbound sessions to always be permitted, use the **sysopt connection permit-ipsec command.**

If both the **sysopt ipsec pl-compatible** command and the **sysopt connection permit-ipsec** command are used within your configuration, the **sysopt ipsec pl-compatible** command will take precedence.

Note The **sysopt ipsec pl-compatible** command is deprecated. In its place, we recommend using the **nat 0 access-list** command to exempt IPSec from NAT.

If the **sysopt connection permit-ipsec** command is not configured, you must explicitly configure an **access-list** command statement to permit IPSec traffic to traverse the PIX Firewall.

The **no sysopt connection permit-ipsec** command disables the option.

sysopt connection permit-pptp

Let PPTP traffic bypass **conduit** and **access-list** command statement checking. Use the **vpdn** command to implement PPTP.

sysopt connection permit-l2tp

This command allows L2TP traffic to bypass conduit or access list checking. Because L2TP traffic can only come from IPSec, the **sysopt connection permit-ipsec** command will allow L2TP traffic to pass as well.

sysopt ipsec pl-compatible

Note The **sysopt ipsec pl-compatible** command provides a migration path for Private Link users from Private Link tunnels to IPSec tunnels.

The **sysopt ipsec pl-compatible** command enables the IPSec feature to simulate the Private Link feature supported in PIX Firewall Version 4. The Private Link feature provides encrypted tunnels to be established across an unsecured network between Private-Link equipped PIX Firewall units. The **sysopt ipsec pl-compatible** command allows IPSec packets to bypass the NAT and ASA features and enables incoming IPSec packets to terminate on the sending interface.

The **sysopt ipsec pl-compatible** command is not available on a PIX 501.

The **no sysopt ipsec pl-compatible** command disables the option, which is off by default.

Note When using the **sysopt ipsec pl-compatible** command, all PIX Firewall features, such as access list control, stateful inspection, and user authentication, are bypassed for IPSec packets only.

If both the **sysopt ipsec pl-compatible** command and the **sysopt connection permit-ipsec** command are used within your configuration, the **sysopt ipsec pl-compatible** command will take precedence.

If the **alias** command is used with the **sysopt ipsec pl-compatible** command, a static **route** command statement must be added for each IP address specified in the **alias** command statement.

sysopt connection tcpmss

The **sysopt connection tcpmss** command allows you to set the minimum and the maximum TCP segment size. Both the host and the server can set the maximum segment size when they first establish a connection. If either maximum exceeds the value you set with the **sysopt connection tcpmss** command, then the PIX firewall overrides the maximum and inserts the value you set. If either maximum is less than the value you set with the **sysopt connection tcpmss minimum** command, then the PIX firewall overrides the maximum and inserts the minimum value you set. For example, if you set a maximum size of 1200 bytes and a minimum size of 400 bytes, when a host requests a maximum size of 1300 bytes, then the PIX firewall alters the packet to request 1200 bytes (the maximum). If another host requests a maximum value of 300 bytes, then the PIX firewall alters the packet to request 400 bytes (the minimum).

The *bytes* value can be a minimum of 48 and any maximum number. You can disable this feature by setting *bytes* to 0. By default, the PIX firewall sets 1380 bytes as the **sysopt connection tcpmss** maximum limit and 48 bytes as the minimum limit, even though this command does not appear in the default configuration. The default of 1380 bytes allows room for header information so that the total packet size does not exceed 1500 bytes, which is the default MTU for Ethernet. See the following calculation:

```
1380 data + 20 TCP + 20 IP + 24 AH + 24 ESP_CIPHER + 12 ESP_AUTH + 20 IP = 1500 bytes
```

If the host or server does not request a maximum segment size, the PIX firewall assumes that the RFC 793 default value of 536 bytes is in effect.

You might want to set the maximum segment size using this command so that the size is less than the MTU and packets are not fragmented. Large numbers of fragments can impact the performance of the PIX firewall when it uses the Frag Guard feature. Setting the minimum size prevents the TCP server from sending many small TCP data packets to the client and impacting the performance of the server and the network.

Note Although, not advised for normal use of this feature, if you encounter the syslog IPFRAG messages 209001 and 209002, you can raise the *bytes* value.

sysopt connection timewait

By default the PIX Firewall does not use the **timewait** option.

Use the **sysopt connection timewait** command to enable the **timewait** option when you have an end host application whose default TCP terminating sequence is a simultaneous close.

This is recommended because the default behavior of the PIX Firewall is to track the shutdown sequence and release the connection after two FINs and the ACK (acknowledgment) of the last FIN segment. This quick release heuristic enables the PIX Firewall to sustain a high connection rate, based on the most common closing sequence, known as the normal close sequence. However, in a simultaneous close, both ends of the transaction initiate the closing sequence, as opposed to the normal close sequence where one end closes and the other end acknowledges prior to initiating its own closing sequence (see RFC 793). Thus, in a simultaneous close, the quick release forces one side of the connection to linger in the CLOSING state. Having many sockets in the CLOSING state can degrade the performance of an end host. For instance, some WinSock mainframe clients are known to exhibit this behavior and degrade the performance of the mainframe server. Old versions of HP/UX are also susceptible to this behavior. Using the **sysopt connection timewait** command creates a window for the simultaneous close down sequence to complete.

The **no sysopt connection timewait** command removes the **sysopt connection timewait** command from your configuration. In other words, if you enable the **timewait** option with the **sysopt connection timewait** command, you can disable it using the **no sysopt connection timewait** command.

Note The **sysopt connection timewait** command requires more system resources than default processing and, when in use, may impact PIX Firewall performance. Noticeable performance impact is most likely when there is limited memory available, and when there is highly dynamic traffic such as HTTP.

sysopt nodnsalias

The **sysopt nodnsalias inbound** disables inbound embedded DNS A record fixups according to aliases that apply to the A record address. **sysopt nodnsalias outbound** affects outbound replies.

This command remedies the case when a DNS server is on the outside and users on the inside need to access a server on a perimeter interface. In the past, you would use the **alias** command to permit DNS responses to resolve correctly through the PIX Firewall, but formerly you had to reverse the parameters for the local IP address and foreign IP address.

For example, you would normally code the **alias** command as follows:

```
alias (inside) 192.168.1.4 209.165.201.11 255.255.255.255
```

Inside host 192.168.1.5 needs access to www.example.com, which resolves at an outside ISP DNS to 209.165.201.11. The PIX Firewall fixes this DNS response sending the host a response of 192.168.1.4. The host uses its gateway (the PIX Firewall) to go to 192.168.1.4, which the PIX Firewall now aliases back to the 209.165.201.11. Because this is actually 192.168.1.4, a server on the perimeter interface of the PIX Firewall, the packet is dropped because the PIX Firewall sent the packet to the outside interface, which is the incorrect interface.

The **sysopt nodnsalias inbound** command has the same effect as reversing the **alias** command statement parameters as follows:

```
alias (inside) 209.165.201.11 192.168.1.4 255.255.255.255
```

This works properly because everything happens in reverse. The DNS is now modified to 209.165.201.11 and the host inside uses its gateway (the PIX Firewall) to get there, the PIX Firewall aliases this back to 192.168.1.4 and routes it out the perimeter interface to the correct host and the TCP connection is established.

sysopt noproxyarp

ARP (Address Resolution Protocol) is a layer two protocol that resolves an IP address to a physical address, also called a Media Access Controller (MAC) address. A host sends an ARP request asking "Who is this IP?" The device owning the IP should reply with "Hey, I am the one, here's my MAC address."

Proxy ARP refers to a gateway device, in this case, the firewall, "impersonating" an IP address and returning its own MAC address to answer an ARP request for another device.

The firewall builds a table from responses to ARP requests to map physical addresses to IP addresses. A periodic ARP function is enabled in the default configuration. The presence of entries in the ARP cache indicates that the firewall has network connectivity. The show arp command lists the entries in the ARP table. Usually, administrators do not need to manually manipulate ARP entries on the firewall. This is done only when troubleshooting or solving network connectivity problems.

The arp command is used to add a permanent entry for host on a network. If one host is exchanged for another host with the same IP address then the "clear arp" command can be used to clear the ARP cache on the PIX. Alternatively, you can wait for the duration specified with the arp timeout command to expire and the ARP table rebuilds itself automatically with the new host information.

The sysopt noproxyarp command is used to disable Proxy ARPs on an interface from the command-line interface. By default, the PIX Firewall responds to ARP requests directed at the PIX Firewall's interface IP addresses as well as to ARP requests for any static or global address defined on the PIX Firewall interface (which are proxy ARP requests).

The **sysopt noproxyarp** *if_name* command lets you disable proxy ARP request responses on a PIX Firewall interface. However, this command does not disable (non-proxy) ARP requests on the PIX Firewall interface itself. Consequently, if you use the **sysopt noproxyarp** *if_name* command, the PIX Firewall no longer responds to ARP requests for the addresses in the **static**, **global**, and **nat 0** commands for that interface but does respond to ARP requests for its interface IP addresses.

To disable Proxy ARPs on the inside interface:

```
sysopt noproxyarp inside
```

To enable Proxy ARPs on the inside interface:

```
no sysopt noproxyarp inside
```

sysopt radius ignore-secret

Some commonly used RADIUS servers, such as Livingston Version 1.16, have a usage caveat where they do not include the key in the authenticator hash in the accounting acknowledgment response. This can cause the PIX Firewall to continually retransmit the accounting request. Use the **sysopt radius ignore-secret** command to cause the PIX Firewall to ignore the key in the authenticator of accounting acknowledgments thus avoiding the retransmit problem. (The key described here is the key you set with the **aaa-server** command.)

show sysopt

The **show sysopt** command lists the **sysopt** commands in the configuration. The **clear sysopt** command resets the **sysopt** command to default settings.

Deprecated Commands

The **sysopt route dnat** and **sysopt security fragguard** commands are deprecated commands.

Examples

The following displays the default sysopt configuration:

```
pixfirewall(config)# show sysopt
no sysopt connection timewait
sysopt connection tcpmss 1380
sysopt connection tcpmss minimum 0
no sysopt nodnsalias inbound
no sysopt nodnsalias outbound
no sysopt radius ignore-secret
no sysopt uauth allow-http-cache
no sysopt connection permit-ipsec
no sysopt connection permit-pptp
no sysopt connection permit-l2tp
no sysopt ipsec pl-compatible
```

In the following example, a PPTP client authenticates using MS-CHAP, negotiates MPPE encryption, receives the DNS and WINS server addresses, and Telnets to the host 192.168.0.2 directly through the **nat 0** command.

```
ip local pool my-addr-pool 10.1.1.1-10.1.1.254
aaa-server my-aaa-server-group (inside) host 192.168.0.10 key 12345678
aaa-server my-aaa-server-group protocol radius
vpdn group 1 accept dialin pptp
vpdn group 1 ppp authentication mschap
vpdn group 1 ppp encryption mppe auto required
vpdn group 1 client configuration address local my-addr-pool
vpdn group 1 client authentication aaa my-aaa-server-group
vpdn group 1 client configuration dns 10.2.2.99
vpdn group 1 client configuration wins 10.2.2.100
vpdn enable outside
access-list nonat permit ip 10.1.1.0 255.255.255.0 host 192.168.0.2
access-list nonat permit ip 10.1.1.0 255.255.255.0 host 10.2.2.99
access-list nonat permit ip 10.1.1.0 255.255.255.0 host 10.2.2.100
nat (inside) 0 access-list nonat
sysopt connection permit-pptp
```

sysopt connection permit-ipsec

The following is a minimal IPSec configuration to enable a session to be connected from host 172.21.100.123 to host 172.21.200.67 across an IPSec tunnel that terminates from peer 209.165.201.1 to peer 201.165.200.225.

With **sysopt connection permit-ipsec** and **access-list** command statements:

On peer 209.165.201.1:

```
static 172.21.100.123 172.21.100.123
access-list 10 permit ip host 172.21.200.67 host 172.21.100.123
crypto ipsec transform-set t1 esp-des esp-md5-hmac
crypto map mymap 10 ipsec-isakmp
crypto map mymap 10 match address 10
crypto map mymap 10 set transform-set t1
crypto map mymap 10 set peer 172.21.200.1
crypto map mymap interface outside
```

On peer 201.165.200.225:

```
static 172.21.200.67 172.21.200.67
access-list 10 permit ip host 172.21.100.123 host 172.21.200.67
crypto ipsec transform-set t1 esp-des esp-md5-hmac
crypto map mymap 10 ipsec-isakmp
crypto map mymap 10 match address 10
crypto map mymap 10 set transform-set t1
crypto map mymap 10 set peer 172.21.100.1
crypto map mymap interface outside
```

With **sysopt connection permit-ipsec** and without **conduit** command statements:

On peer 209.165.201.1:

```
static 172.21.100.123 172.21.100.123
access-list 10 permit ip host 172.21.200.67 host 172.21.100.123
crypto ipsec transform-set t1 esp-des esp-md5-hmac
crypto map mymap 10 ipsec-isakmp
crypto map mymap 10 match address 10
crypto map mymap 10 set transform-set t1
crypto map mymap 10 set peer 172.21.200.1
crypto map mymap interface outside
sysopt connection permit-ipsec
```

On peer 201.165.200.225:

```
static 172.21.200.67 172.21.200.67
access-list 10 permit ip host 172.21.100.123 host 172.21.200.67
crypto ipsec transform-set t1 esp-des esp-md5-hmac
crypto map mymap 10 ipsec-isakmp
crypto map mymap 10 match address 10
crypto map mymap 10 set transform-set t1
crypto map mymap 10 set peer 172.21.100.1
crypto map mymap interface outside
sysopt connection permit-ipsec
```

CHAPTER 9

T through Z Commands

telnet

Specify the host for PIX Firewall console access via Telnet.

telnet *ip_address* [*netmask*] [*if_name*]

clear telnet [*ip_address* [*netmask*] [*if_name*]]

no telnet [*ip_address* [*netmask*] [*if_name*]]

telnet timeout *minutes*

show telnet

show telnet timeout

Syntax Description

if_name	If IPSec is operating, PIX Firewall lets you specify an unsecure interface name, typically, the outside interface. At a minimum, the **crypto map** command must be configured to specify an interface name with the **telnet** command.
ip_address	An IP address of a host or network that can access a PIX Firewall Telnet management session. If an interface name is not specified, the address is assumed to be on an internal interface. PIX Firewall automatically verifies the IP address against the IP addresses specified by the **ip address** commands to ensure that the address you specify is on an internal interface. If an interface name is specified, PIX Firewall only checks the host against the interface you specify.
netmask	Bit mask of *ip_address*. To limit access to a single IP address, use 255 in each octet; for example, 255.255.255.255. If you do not specify *netmask*, it defaults to 255.255.255.255 regardless of the class of *local_ip*. Do not use the subnetwork mask of the internal network. The *netmask* is only a bit mask for the IP address in *ip_address*.
timeout *minutes*	The number of minutes that a Telnet session can be idle before being closed by PIX Firewall. The default is 5 minutes. The range is **1** to **60** minutes.

Command Modes

Configuration mode.

Usage Guidelines

The **telnet** command lets you specify which hosts can access the PIX Firewall console with Telnet. You can enable Telnet to the PIX Firewall on all interfaces. However, the PIX Firewall enforces that all Telnet traffic to the outside interface be IPSec protected. Therefore, to enable Telnet session to the outside interface, configure IPSec on the outside interface to include IP traffic generated by the PIX Firewall and enable Telnet on the outside interface.

A maximum of five (5) active Telnet management sessions to the PIX Firewall are allowed at the same time. The **show telnet** command displays the current list of IP addresses authorized to Telnet to the PIX Firewall. Use the **no telnet** or **clear telnet** command to remove Telnet access from a previously set IP address. Use the **telnet timeout** feature to set the maximum time a console Telnet session can be idle before being logged off by PIX Firewall. The **clear telnet** command does not affect the **telnet timeout** command duration. The **no telnet** command cannot be used with the **telnet timeout** command.

Use the **passwd** command to set a password for Telnet access to the console. The default is **cisco**. Use the **who** command to view which IP addresses are currently accessing the PIX Firewall console. Use the **kill** command to terminate an active Telnet management session.

If the **aaa** command is used with the **console** option, Telnet management access must be authenticated with an authentication server.

Note If you have configured the **aaa** command to require authentication for PIX Firewall Telnet management access and the console login request times out, you can gain access to the PIX Firewall from the serial console by entering the **pix** username and the password that was set with the **enable password** command.

Usage Notes

1. If you do not specify the interface name, the **telnet** command adds command statements to the configuration to let the host or network access the Telnet management session from all internal interfaces.

   When you use the **show telnet** command, this assumption may not seem to make sense. For example, if you enter the following command without a netmask or interface name.

   ```
   telnet 192.168.1.1
   ```

   If you then use the **show telnet** command, you see that not just one command statement is specified, but all internal interfaces are represented with a command statement:

   ```
   show telnet
   192.168.1.1 255.255.255.255 inside
   192.168.1.1 255.255.255.255 intf2
   192.168.1.1 255.255.255.255 intf3
   ```

   The purpose of the **show telnet** command is that, were it possible, the 192.168.1.1 host could access the Telnet management session from any of these internal interfaces. An additional facet of this behavior is that you must delete each of these command statements individually with the following commands.

   ```
   no telnet 192.168.1.1 255.255.255.255 inside
   no telnet 192.168.1.1 255.255.255.255 intf2
   no telnet 192.168.1.1 255.255.255.255 intf3
   ```

2. To access the PIX Firewall with Telnet from the intf2 perimeter interface, use the following command:

   ```
   telnet 192.168.1.1 255.255.255.255 int2
   ```

3. The default password to access the PIX Firewall console via Telnet is **cisco**.

4. Some Telnet applications such as the Windows 95 or Windows NT Telnet sessions may not support access to the PIX Firewall unit's command history feature via the arrow keys. However, you can access the last entered command by pressing Ctrl-P.
5. The **telnet timeout** command affects the next session started but not the current session.
6. If you connect a computer directly to the inside interface of the PIX Firewall with Ethernet to test Telnet access, you must use a cross-over cable and the computer must have an IP address on the same subnet as the inside interface. The computer must also have its default route set to be the inside interface of the PIX Firewall.
7. If you need to access the PIX Firewall console from outside the PIX Firewall, you can use a **static** and **access-list** command pair to permit a Telnet session to a Telnet server on the inside interface, and then from the server to the PIX Firewall. In addition, you can attach the console port to a modem but this may add a security problem of its own. You can use the same terminal settings as for HyperTerminal, which is described in the *Cisco PIX Firewall and VPN Configuration Guide*.

   If you have IPSec configured, you can access the PIX Firewall console with Telnet from outside the PIX Firewall. Once an IPSec tunnel is created from an outside host to the PIX Firewall, you can access the console from the outside host.
8. Output from the **debug crypto ipsec**, **debug crypto isakmp**, and **debug ssh** commands do not display in a Telnet or SSH console session. For information about the **debug crypto ipsec** and **debug crypto isakmp** commands, refer to the **debug** command page.

Examples

The following examples permit hosts 192.168.1.3 and 192.168.1.4 to access the PIX Firewall console via Telnet. In addition, all the hosts on the 192.168.2.0 network are given access:

```
telnet 192.168.1.3 255.255.255.255 inside
telnet 192.168.1.4 255.255.255.255 inside
telnet 192.168.2.0 255.255.255.0 inside
show telnet
        192.168.1.3 255.255.255.255 inside
        192.168.1.4 255.255.255.255 inside
        192.168.2.0 255.255.255.0 inside
```

You can remove individual entries with the **no telnet** command or all **telnet** command statements with the **clear telnet** command:

```
no telnet 192.168.1.3 255.255.255.255 inside
show telnet
        192.168.1.4 255.255.255.255 inside
        192.168.2.0 255.255.255.0 inside
clear telnet
show telnet
```

You can change the maximum session idle duration as follows:

```
telnet timeout 10
show telnet timeout
telnet timeout 10 minutes
```

An example Telnet login session appears as follows (the password does not display when entered):

```
PIX passwd: cisco

Welcome to the PIX Firewall
…
Type help or '?' for a list of available commands.
pixfirewall>
```

Related Commands

- aaa accounting
- kill
- **password**
- **who**

terminal

Change console terminal settings.

terminal monitor

terminal no monitor

terminal width *characters*

Syntax Description

characters	Permissible values are 0, which means 511 characters, or a value in the range of 40 to 511.
monitor	Enable or disable syslog message displays on the console.
width	Set the width for displaying information during console sessions.

Command Modes

Configuration mode.

Usage Guidelines

The **terminal monitor** command lets you enable or disable the display of syslog messages in the current session for either Telnet or serial access to the PIX Firewall console. Use the **logging monitor** command to enable or disable various levels of syslog messages to the console; use the **terminal no monitor** command to disable the messages on a per session basis. Use **terminal monitor** to restart the syslog messages for the current session.

The **terminal width** command sets the width for displaying command output. The terminal width is controlled by the command: **terminal width** *nn*, where *nn* is the width in characters. If you enter a line break, it is not possible to backspace to the previous line.

Examples

The following example shows enabling logging and then disabling logging only in the current session with the **terminal no monitor** command:

```
logging monitor
…
terminal no monitor
```

tftp-server

Specify the IP address of the TFTP configuration server.

[no] tftp-server [*if_name*] *ip_address path*

clear tftp-server [[*if_name*] *ip_address path*]

show tftp-server

Syntax Description

if_name	Interface name on which the TFTP server resides. If not specified, an internal interface is assumed. If you specify the outside interface, a warning message informs you that the outside interface is unsecure.
ip_address	The IP address or network of the TFTP server.
path	The path and filename of the configuration file. The format for path differs by the type of operating system on the server. The contents of path are passed directly to the server without interpretation or checking. The configuration file must exist on the TFTP server. Many TFTP servers require the configuration file to be world-writable to write to it and world-readable to read from it.

Command Modes

Configuration mode.

Usage Guidelines

The **tftp-server** command lets you specify the IP address of the server that you use to propagate PIX Firewall configuration files to your firewalls. Use the **tftp-server** command with the **configure net** command to read from the configuration or with the **write net** command to store the configuration in the file you specify. The **clear tftp-server** command removes the **tftp-server** command from your configuration.

PIX Firewall supports only one TFTP server.

The *path* name you specify in the **tftp-server** is appended to the end of the IP address you specify in the **configure net** and **write net** commands. The more you specify of a file and path name with the **tftp-server** command, the less you need to specify with the **configure net** and **write net** commands. If you specify the full path and filename in the **tftp-server** command, the IP address in the **configure net** and **write net** commands can be represented with a colon (**:**).

The **no tftp server** command disables access to the server. The **show tftp-server** command lists the **tftp-server** command statements in the current configuration.

Note If the TFTP server to which the firewall is trying to connect is not running the TFTP service, the firewall hangs and does not timeout. Press "ESC" key on the firewall console to abort the TFTP session and return to the firewall command line prompt.

Examples

The following example specifies a TFTP server and then reads the configuration from /pixfirewall/config/test_config:

```
tftp-server 10.1.1.42 /pixfirewall/config/test_config
...
```

```
configure net :
```

timeout

Set the maximum idle time duration.

timeout [**xlate** *hh*[**:***mm*[**:***ss*]]] [**conn** *hh*[**:***mm*[**:***ss*]]] [**half-closed** *hh*[**:***mm*[**:***ss*]]] [**udp** *hh*[**:***mm*[**:***ss*]]] [**rpc** *hh*[**:***mm*[**:***ss*]]] [**h225** *hh*[**:***mm*[**:***ss*]]] [**h323** *hh*[**:***mm*[**:***ss*]]] [**mgcp** *hh*[**:***mm*[**:***ss*]]] [**sip** *hh*[**:***mm*[**:***ss*]]] [**sip_media** *hh*[**:***mm*[**:***ss*]]][**uauth** *hh*[**:***mm*[**:***ss*]] [**absolute** | **inactivity**]]

clear timeout

show timeout

Syntax Description

absolute	Run **uauth** timer continuously, but after timer elapses, wait to reprompt the user until the user starts a new connection, such as clicking a link in a web browser. The default **uauth** timer is **absolute**. To disable **absolute**, set the uauth timer to **0** (zero).
conn *hh*[**:***mm*[**:***ss*]]	Idle time after which a connection closes. Use **0:0:0** for the time value to never time out a connection. This duration must be at least 5 minutes. The default is 1 hour.
h225 *hh*[**:***mm*[**:***ss*]]	The idle time after which H.225 signalling closes, where *hh* is hours, *mm* is minutes, and *ss* is seconds. The default is 1 hour. A timeout value of **h225 00:00:00** means never tear down H.225 signalling. A timeout value of **h225 00:00:01** disables the timer and closes the TCP connection immediately after all calls are cleared.
h323 *hh*[**:***mm*[**:***ss*]]	The idle time after which an H.323 media connection closes. The default is 5 minutes. (This is the H.323 UDP inactivity timer.)
half-closed *hh*[**:***mm*[**:***ss*]]	Idle time until a TCP half-close connection is freed. The default is 10 minutes. Use **0:0:0** to never time out a half-closed connection. The minimum is 5 minutes.
inactivity	Start **uauth** timer after a connection becomes idle.
mgcp *hh*[**:***mm*[**:***ss*]]	Sets the duration for the Media Gateway Control Protocol (MGCP) inactivity timer. The default is 5 minutes.
rpc *hh*[**:***mm*[**:***ss*]]	Idle time until an RPC slot is freed. This duration must be at least 1 minute. The default is 10 minutes.
sip *hh*[**:***mm*[**:***ss*]]	Modifies the SIP timer which is used for UDP signalling connections identified by the value T in the output from the **show conn detail** command. The default timeout value is 30 seconds.
sip_media *hh*[**:***mm*[**:***ss*]]	Modifies the media timer, which is used for SIP RTP/RTCP with SIP UDP media packets, instead of the UDP inactivity timeout. SIP media port is set to 2 minutes in the list of protocol timers.
uauth *hh*[**:***mm*[**:***ss*]]	Duration before authentication and authorization cache times out and user has to re authenticate next connection. This duration must be shorter than the **xlate** values. Set to **0** to disable caching. Do not set to zero if passive FTP is used on the connections.

udp *hh*[**:***mm*[**:***ss*]]	Idle time until a UDP slot is freed. This duration must be at least 1 minute. The default is 2 minutes.
xlate *hh*[**:***mm*[**:***ss*]]	Idle time until a translation slot is freed. This duration must be at least 1 minute. The default is 3 hours. **Note** PIX Firewall clears UDP PAT connections 30 seconds after the connection is closed, regardless of the setting of the **timeout xlate** command.

Command Modes

Configuration mode.

Usage Guidelines

The **timeout** command sets the idle time for connection, translation UDP, RPC, and H.323 slots. If the slot has not been used for the idle time specified, the resource is returned to the free pool. TCP connection slots are freed approximately 60 seconds after a normal connection close sequence.

The **clear timeout** command sets the durations to their default values.

This command is used in conjunction with the **show** and **clear uauth** commands.

Note Do not use the **timeout uauth 0:0:0** command if passive FTP is used for the connection, or if the **virtual** command is used for Web authentication.

The connection timer takes precedence over the translation timer, such that the translation timer only works after all connections have timed out.

timeout mgcp

The **timeout mgcp** *hh:mm:ss* command sets the duration for the MGCP inactivity timer. If this time elapses before new activity occurs, the MGCP media ports close. The default is five minutes. For example, to set the MGCP timeout to five minutes, enter the following:

```
pixfirewall(config)# timeout mgcp 00:05:00
```

Uauth Inactivity and Absolute Qualifiers

The **uauth inactivity** and **absolute** qualifiers cause users to have to reauthenticate after either a period of inactivity or an absolute duration.

If you set the inactivity timer to a duration, but the absolute timer to zero, then users are only reauthenticated after the inactivity timer elapses. If you set both timers to zero, then users have to reauthenticate on every new connection.

The inactivity timer starts after a connection becomes idle. If a user establishes a new connection before the duration of the inactivity timer, the user is not required to reauthenticate. If a user establishes a new connection after the inactivity timer expires, the user must reauthenticate. The default durations are zero for the inactivity timer and 5 minutes for the absolute timer; that is, the default behavior is to cause the user to reauthenticate every 5 minutes.

The absolute timer runs continuously, but waits to reprompt the user when the user starts a new connection, such as clicking a link and the absolute timer has elapsed, then the user is prompted to reauthenticate. The absolute timer must be shorter than the **xlate** timer; otherwise, a user could be reprompt after their session already ended.

Inactivity timers give users the best Web access because they are not prompted to regularly reauthenticate. Absolute timers provide security and manage the PIX Firewall connections better. By being prompted to reauthenticate regularly, users manage their use of the resources more efficiently. Also by being reprompted, you minimize the risk that someone will attempt to use another user's access after they leave their workstation, such as in a college computer lab. You may want to set an absolute timer during peak hours and an inactivity timer thereafter.

Both an inactivity timer and an absolute timer can operate at the same time, but you should set the absolute timer duration longer than the inactivity timer. If the absolute timer is less than the inactivity timer, the inactivity timer never occurs. For example, if you set the absolute timer to 10 minutes and the inactivity timer to an hour, the absolute timer reprompts the user every 10 minutes; therefore, the inactivity timer will never be started.

Note RPC and NFS are very unsecure protocols and should be used with caution.

Examples

The following is sample output from the **show timeout** command:

```
show timeout
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h323 0:05:00
sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
```

The following is sample output from the **timeout** command in which variables are changed and then displayed with the **show timeout** command:

```
timeout uauth 0:5:00 absolute uauth 0:4:00 inactivity
show timeout
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h323 0:05:00
sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute uauth 0:04:00 inactivity
```

Related Commands

- **show xlate/clear xlate**
- **show uauth/clear uauth**

url-block

For Websense filtering servers, the **url-block url-size** command allows filtering of long URLs, up to 4 KB. For both Websense and N2H2 filtering servers, the **url-block block** command causes the PIX Firewall to buffer packets received from a web server in response to a web client request while waiting for a response from the URL filtering server. This improves performance for the web client compared to the default PIX Firewall behavior, which is to drop the packets and to require the web server to retransmit the packets if the connection is permitted.

If you use the **url-block block** command and the filtering server permits the connection, the PIX Firewall sends the blocks to the web client from the HTTP response buffer and removes the blocks from the buffer. If the filtering server denies the connection, the PIX Firewall sends a deny message to the web client and removes the blocks from the HTTP response buffer.

[**no**] **url-block block** *block_buffer_limit*

clear url-block block stat

show url-block block stat

Websense only:

[**no**] **url-block url-mempool** *memory_pool_size*

[**no**] **url-block url-size** *long_url_size*

Syntax Description

block *block_buffer_limit*	Creates an HTTP response buffer to store web server responses while waiting for a filtering decision from the filtering server. The permitted values are from 0 to 128, with specifies the number of 1550-byte blocks.
stat	Displays block buffer usage statistics.
url-mempool *memory_pool_size*	For Websense URL filtering only. The size of the URL buffer memory pool in Kilobytes (KB). The permitted values are from 2 to 10240, which specifies a URL buffer memory pool from 2 KB to 10240 KB.
url-size *long_url_size*	For Websense URL filtering only. The maximum allowed URL size in KB. The permitted values are 2, 3, or 4, which specifies a maximum URL size of 2 KB, 3 KB, or 4KB.

Command Modes

Configuration mode.

Usage Guidelines

Use the url-block block command to specify the number of blocks to use for buffering web server responses while waiting for a filtering decision from the filtering server.

Use the **url-block url-size** command with the **url-block url-mempool** command to specify the maximum length of a URL to be filtered by a Websense filtering server and the maximum memory to assign to the URL buffer. Use these commands to pass URLs longer than 1159 bytes, up to a maximum of 4096 bytes, to the Websense server. The **url-block url-size** command stores URLs longer than 1159 bytes in a buffer and then passes the URL to the Websense server (through a TCP packet stream) so that the Websense server can grant or deny access to that URL.

The **clear url-block block stat** command clears the block buffer usage counters, except for the `Current number of packets held (global)` counter.

The **show url-block block stat** command displays the number of packets held in the url-block buffer and the number (if any) dropped due to exceeding the buffer limit or retransmission.

Examples

The following example illustrates the use of the **show url-block block stat** and **clear url-block block stat** commands:

```
pixfirewall(config)# sh url-block block stat

URL Pending Packet Buffer Stats with max block  128
-----------------------------------------------------
Cumulative number of packets held:              896
Maximum number of packets held (per URL):       3
Current number of packets held (global):        38
Packets dropped due to
       exceeding url-block buffer limit:        7546
       HTTP server retransmission:              10
Number of packets released back to client:      0

pixfirewall(config)# sh url-block
    url-block url-mempool 128
    url-block url-size 4
    url-block block 128

pixfirewall(config)# clear url-block block stat
pixdocipsec1(config)# show url-block block stat

URL Pending Packet Buffer Stats with max block  0
-----------------------------------------------------
Cumulative number of packets held:              0
Maximum number of packets held (per URL):       0
Current number of packets held (global):        38
Packets dropped due to
       exceeding url-block buffer limit:        0
       HTTP server retransmission:              0
Number of packets released back to client:      0
```

url-cache

Caches URL access privileges that were previously retrieved from a Websense or N2H2 server.

[**no**] **url-cache** {**dst** | **src_dst**} **size** *kbytes*

clear url-cache

show url-cache stats

Syntax Description

dst	Cache entries based on the URL destination address. Select this mode if all users share the same URL filtering policy on the N2H2 or Websense server.
size *kbytes*	Specifies a value for the cache size within the range 1 to 128 KB.
src_dst	Cache entries based on the both the source address initiating the URL request as well as the URL destination address. Select this mode if users do not share the same URL filtering policy on the N2H2 or Websense server.
stat	Use the **stat** option to display additional URL cache statistics, including the number of cache lookups and hit rate.

Command Modes

Configuration mode.

Usage Guidelines

The **url-cache** command provides a configuration option to allow the PIX to cache previously retrieved URL access privileges from a Websense or N2H2 server.

Use the **url-cache** command to enable URL caching, set the size of the cache, and display cache statistics.

Caching stores URL access privileges in memory on the PIX Firewall. When a host requests a connection, the PIX Firewall first looks in the URL cache for matching access privileges instead of forwarding the request to the N2H2 or Websense server. Disable caching with the **no url-cache** command.

The **clear url-cache** command removes **url-cache** command statements from the configuration.

Using the URL cache does not update the Websense accounting logs for Websense protocol Version 1. If you are using Websense protocol Version 1, let Websense run to accumulate logs so you can view the Websense accounting information. After you get a usage profile that meets your security needs, enable **url-cache** to increase throughput. Accounting logs are updated for Websense protocol Version 4 and for N2H2 URL filtering while using the **url-cache** command.

Note If you change settings on the N2H2 or Websense server, disable the cache with the **no url-cache** command and then reenable the cache with the **url-cache** command.

The **show url-cache** command with the **stats** option displays the following entries:

- Size—The size of the cache in kilobytes, set with the **url-cache** *size* option.
- Entries—The maximum number of cache entries based on the cache size.
- In Use—The current number of entries in the cache.
- Lookups—The number of times the PIX Firewall has looked for a cache entry.
- Hits—The number of times the PIX Firewall has found an entry in the cache.

You can view additional information about N2H2 or Websense filtering acitivity with the **show perfmon** command.

Examples

The following example caches all outbound HTTP connections based on the source and destination addresses:

```
url-cache src_dst 128
```

The following is sample output from the **show url-cache stat** command:

```
show url-cache stat

URL Filter Cache Stats
----------------------
    Size :     1KB
 Entries :      36
  In Use :      30
 Lookups :      300
    Hits :      290
```

url-server

Designate a server running either N2H2 or Websense for use with the **filter** command; you cannot run both of these URL filtering services simultaneously.

N2H2

[**no**] **url-server** [(*if_name*)] **vendor n2h2 host** *local_ip* [**port** *number*] [**timeout** *seconds*] [**protocol** {**TCP** | **UDP**}]

Websense

[**no**] **url-server** [(*if_name*)] **vendor websense host** *local_ip* [**timeout** *seconds*] [**protocol** {**TCP** | **UDP**} *version*]

show url-server

show url-server stats

Syntax Description

N2H2

host *local_ip*	The server that runs the URL filtering application.
if_name	The network interface where the authentication server resides. If not specified, the default is inside.
port *number*	The N2H2 server port. The PIX Firewall also listens for UDP replies on this port. The default port number is 4005.
protocol	The protocol can be configured using **TCP** or **UDP** keywords. The default is TCP.
timeout *seconds*	The maximum idle time permitted before PIX Firewall switches to the next server you specified. The default is 5 seconds.
vendor n2h2	Indicates URL filtering service vendor is N2H2.

Websense

if_name	The network interface where the authentication server resides. If not specified, the default is inside.
host *local_ip*	The server that runs the URL filtering application.
timeout *seconds*	The maximum idle time permitted before PIX Firewall switches to the next server you specified. The default is 5 seconds.
protocol	The protocol can be configured using **TCP** or **UDP** keywords. The default is TCP protocol, Version 1.
vendor websense	Indicates URL filtering service vendor is Websense.
version	Specifies protocol Version **1** or **4**. The default is TCP protocol Version 1. TCP can be configured using Version 1 or Version 4. UDP can be configured using Version 4 only.

Command Modes

Configuration mode.

Usage Guidelines

The **url-server** command designates the server running the N2H2 or Websense URL filtering application. The limit is 16 URL servers; however, and you can use only one application at a time, either N2H2 or Websense. Additionally, changing your configuration on the PIX Firewall does not update the configuration on the application server; this must be done separately, according to the individual vendor's instructions.

Once you designate the server, enable the URL filtering service with the **filter** command.

Follow these steps to filter URLs:

Step 1 Designate the URL filtering application server with the appropriate form of the vendor-specific **url-server** command.

Step 2 Enable URL filtering with the **filter** command.

Step 3 (Optional) Use the **url-cache** command to enable URL caching to improve perceived response time.

Step 4 (Optional) Enable long URL and HTTP buffering support using the **url-block** commands.

Step 5 Use the **show url-block block stats**, **show url-cache stats**, **show url-server stats**, and the **show pdm** commands to view run information.

For more information about Filtering by N2H2, visit N2H2's website at:

http://www.n2h2.com

For more information on Websense filtering services, visit the following website:

http://www.websense.com/

The **url-server** command must be configured before issuing the **filter** command for HTTPS and FTP. If all URL servers are removed from the server list, then all **filter** commands related to URL filtering are also removed.

show url-server commands

The **show url-server stats** command displays the URL server vendor; number of URLs total, allowed, and denied; number of HTTPS connections total, allowed, and denied; number of TCP connections total, allowed, and denied; and the URL server status.

The **show url-server** command displays the following information:

- For N2H2, **url-server** (*if_name*) **vendor n2h2 host** *local_ip* **port** *number* **timeout** *seconds* **protocol** [{**TCP** | **UDP**}{**version 1** | **4**}]
- For Websense, **url-server** (*if_name*) **vendor websense host** *local_ip* **timeout** *seconds* **protocol** [{**TCP** | **UDP**}]

Examples

Using N2H2, the following example filters all outbound HTTP connections except those from the 10.0.2.54 host:

```
url-server (perimeter) vendor n2h2 host 10.0.1.1
filter url http 0 0 0 0
filter url except 10.0.2.54 255.255.255.255 0 0
```

Using Websense, the following example filters all outbound HTTP connections except those from the 10.0.2.54 host:

```
url-server (perimeter) vendor websense host 10.0.1.1
filter url http 0 0 0 0
filter url except 10.0.2.54 255.255.255.255 0 0
```

The following is sample output from the **show url-server stats** command:

```
pixfirewall# show url-server stats

URL Server Statistics:
----------------------
Vendor websense
HTTPs total/allowed/denied 0/0/0
HTTPSs total/allowed/denied 0/0/0
FTPs total/allowed/denied 0/0/0


URL Server Status:
------------------
172.23.58.103 UP

URL Packets Send and Recieve Stats:
-----------------------------------
Message Send Recieve
STATUS_REQUEST 200 200
LOOKUP_REQUEST 10 10
LOG_REQUEST 20 NA
```

Related Commands

- aaa authorization
- filter
- show

username

Sets the username for the specified privilege level.

username *username {*[{**nopassword | password** *password*} [**encrypted**]] [**privilege** *level*]*}*

no username *username*

clear username

show username *username*

Syntax Description

username	Specifies the name of a specific user in the local PIX Firewall authentication database.

Command Modes

Configuration mode.

Usage Guidelines

The local PIX Firewall user authentication database consists of the users entered with the **username** command. The PIX Firewall **login** command uses this database for authentication.

The **show username** *username* command displays users entered in the local PIX Firewall user authentication database.

Related Commands

- login
- privilege

virtual

Access the PIX Firewall virtual server.

virtual http *ip_address* [**warn**]

virtual telnet *ip_address*

Syntax Description

ip_address	For outbound use, *ip_address* must be an address routed to the PIX Firewall. Use an RFC 1918 address that is not in use on any interface. For inbound use, *ip_address* must be an unused global address. An **access-list** and **static** command pair must provide access to *ip_address*, as well as an **aaa accounting authentication** command statement. See the "Examples" section for more information. For example, if an inside client at 192.168.0.100 has a default gateway set to the inside interface of the PIX Firewall at 192.168.0.1, the *ip_address* can be any IP address not in use on that segment (such as 10.2.3.4). As another example, if the inside client at 192.168.0.100 has a default gateway other than the PIX Firewall (such as a router at 192.168.0.254), then the *ip_address* would need to be set to a value that would get statically routed to the PIX Firewall. This might be accomplished by using a value of 10.0.0.1 for the *ip_address*, then on the client, setting the PIX Firewall at 192.168.0.1 as the route to host 10.0.0.1.
warn	Let **virtual http** command users know that the command was redirected. This option is only applicable for text-based browsers where the redirect cannot happen automatically.

Command Modes

Configuration mode.

Usage Guidelines

The **virtual http** command lets web browsers work correctly with the PIX Firewall **aaa** command. The **aaa** command assumes that the AAA server database is shared with a web server. PIX Firewall automatically provides the AAA server and web server with the same information. The **virtual http** command works with the **aaa** command to authenticate the user, separate the AAA server information from the web client's URL request, and direct the web client to the web server. Use the **show virtual http** command to list commands in the configuration. Use the **no virtual http** command to disable its use.

The **virtual http** command works by redirecting the web browser's initial connection to the *ip_address,* which resides in the PIX Firewall, authenticating the user, then redirecting the browser back to the URL which the user originally requested. This mechanism comprises the PIX Firewall unit's new virtual server feature. The reason this command is named as it is, is because the **virtual http** command accesses the virtual server for use with HTTP, another name for the Web. This command is especially useful for PIX Firewall interoperability with Microsoft IIS, but is useful for other authentication servers.

When using HTTP authentication to a site running Microsoft IIS that has "Basic text authentication" or "NT Challenge" enabled, users may be denied access from the Microsoft IIS server. This occurs because the browser appends the string: "Authorization: Basic=Uuhjksdkfhk==" to the HTTP GET commands. This string contains the PIX Firewall authentication credentials.

Windows NT Microsoft IIS servers respond to the credentials and assume that a Windows NT user is trying to access privileged pages on the server. Unless the PIX Firewall username password combination is exactly the same as a valid Windows NT username and password combination on the Microsoft IIS server, the HTTP GET command is denied.

To solve this problem, PIX Firewall provides the **virtual http** command which redirects the browser's initial connection to another IP address, authenticates the user, then redirects the browser back to the URL which the user originally requested.

Once authenticated, a user never has to reauthenticate no matter how low the PIX Firewall uauth timeout is set. This is because the browser caches the "Authorization: Basic=Uuhjksdkfhk==" string in every subsequent connection to that particular site. This can *only* be cleared when the user exits *all* instances of Netscape Navigator or Internet Explorer and restarts. Flushing the cache is of no use.

If you want double authentication through the authentication and web browser, configure the authentication server to not accept anonymous connections.

Note Do not set the **timeout uauth** duration to 0 seconds when using the **virtual** command because this will prevent HTTP connections to the real web server.

For both the **virtual http** and **virtual telnet** commands, if the connection is started on either an outside or perimeter interface, a **static** and **access-list** command pair is required for the fictitious IP address.

The **virtual telnet** command allows the Virtual Telnet server to provide a way to pre-authenticate users who require connections through the PIX Firewall using services or protocols that do not support authentication.

The **virtual telnet** command can be used both to log in and log out of the PIX Firewall. When an unauthenticated user Telnets to the virtual IP address, they are challenged for their username and password, and then authenticated with the TACACS+ or RADIUS server. Once authenticated, they see the message "Authentication Successful" and their authentication credentials are cached in the PIX Firewall for the duration of the uauth timeout.

If a user wishes to log out and clear their entry in the PIX Firewall uauth cache, the user can again Telnet to the virtual address. The user is prompted for their username and password, the PIX Firewall removes the associated credentials from the uauth cache, and the user will receive a "Logout Successful" message.

If inbound users on either the perimeter or outside interfaces need access to the Virtual Telnet server, a **static** and **access-list** command pair must accompany use of the **virtual telnet** command.

The Virtual Telnet server provides a way to pre-authenticate users who require connections through the PIX Firewall using services or protocols that do not support authentication. Users first connect to the Virtual Telnet server IP address, where the user is prompted for a username and password.

Examples

- **virtual http**—The following example shows the commands required to use the **virtual http** command for an inbound connection:

```
static (inside, outside) 209.165.201.1 209.165.201.1 netmask 255.255.255.255
access-list acl_out permit tcp any host 209.165.201.1 eq 80
access-group acl_out in interface outside
aaa authentication include any inbound 209.165.201.1 255.255.255.255 0 0 tacacs+
virtual http 209.165.201.1
```

  This configuration uses an identity static, where both the global IP address and the local address in the static command is the IP address of the virtual server.

  The next example is sample output from the **show virtual** command:

```
show virtual http
virtual http 209.165.201.1
```

- **virtual telnet**—After adding the **virtual telnet** command to the configuration and writing the configuration to Flash memory, users wanting to start PPTP sessions through PIX Firewall use Telnet to access the *ip_address* as shown in the following example:

  On the PIX Firewall:

```
virtual telnet 209.165.201.25
static (inside, outside) 209.165.201.25 209.165.201.25 netmask 255.255.255.255
access-list acl_out permit tcp any host 209.165.201.25 eq telnet
access-group acl_out in interface outside
write memory
```

  This configuration uses an identity static, where both the global IP address and the local address in the static command is the IP address of the virtual server.

  On an inside host:

```
/unix/host%telnet 209.165.201.30
Trying 209.165.201.25...
Connected to 209.165.201.25.
Escape character is '^]'.

username: username

TACACS+ Password: password

Authentication Successful

Connection closed by foreign host.
/unix/host%
```

  The *username* and *password* are those for the user on the TACACS+ server.

vpdn

Configure Virtual Private Dial-up Networking using the L2TP, PPTP, or PPPoE.

vpdn group *group_name* [[**accept dialin pptp | l2tp**] | **request dialout pppoe**] | [**ppp authentication pap|chap|mschap**] | [**ppp encryption mppe 40 | 128| auto** [**required**]] | [**client configuration address local** *address_pool_name*] | [**client configuration dns** *dns_ip1* [*dns_ip2*]] | [**client configuration wins** *wins_ip1* [*wins_ip2*]] | [**client authentication local | aaa** *auth_aaa_group*] | [**client accounting** *acct_aaa_group*] | [**pptp echo** *echo_time*] | [**l2tp tunnel hello** *hello_time*]

vpdn username *name* **password** *passwd* [**store-local**]

vpdn enable *if_name*

show vpdn tunnel [**l2tp|pptp|pppoe**] [**id** *tnl_id* | **packets** | **state** | **summary** | **transport**]

show vpdn session [**l2tp|pptp|pppoe**] [**id** *sess_id* | **packets** | **state| window**]

show vpdn pppinterface [**id** *dev_id*]

show vpdn group [*group_name*]

show vpdn username [**user_name**]

clear vpdn [**group** | **interface| tunnel** *tnl_id* | **username**]

Syntax Description

accept dialin pptp|l2tp **pptp**	Accept a dial-in request using PPTP or L2TP.
all	[**clear** command only]—Removes all L2TP or PPTP tunnels from the configuration.
client accounting aaa-server-group	Specifies the AAA server group for accounting. The accounting AAA server group can be different from the AAA server group for user authentication.
client authentication aaa *aaa_server_group*	Specifies the AAA server group for user authentication.
client authentication local	Authenticate using the local username and password entries you specify in the PIX Firewall configuration.
client configuration address local *address_pool_name*	Specifies the local address pool used to allocate an IP address to a client. Use the **ip local pool** command to specify the IP addresses for use by the clients.
client configuration dns *dns_server_ip1* [*dns_server_ip2*]	Specifies up to two DNS server IP addresses. If set, the PIX Firewall sends this information to the Windows client during the IPCP phase of PPP negotiation.
client configuration wins *wins_server_ip1* [*wins_server_ip2*]	Specifies up to two WINS server IP addresses.
enable *if_name*	Enable the VPDN function on a PIX Firewall interface. Specifies the interface in *if_name* where L2TP or PPTP traffic is received. Only inbound connections are supported.

group	[**clear** command only]—Removes all **vpdn group** commands from the configuration.
group *group_name*	Specifies the VPDN group name. The VPDN *group_name* is an ASCII string to denote a VPDN group. You can make up the name. The maximum length is 63 characters.
id	Identify tunnel or session.
id *session_id*	Unique session identifier.
id *tnl_id*	Unique tunnel identifier.
l2tp | *pptp* | **pppoe**	Select either *l2tp, pptp*, or **pppoe** to display information for only that tunnel type.
l2tp tunnel hello *hello_timeout*	Specifies L2TP tunnel keep-alive hello timeout value in seconds. Default is 60 seconds if not specified. The value can be between10 to 300 seconds.
localname *username*	Assigns a name to the group for PPPoE use. This is also the *name* in the **vpdn username** command.
packets	Packet and byte count.
passwd	Specifies the password for the local group used for PPPoE.
password	Specifies local user password.
ppp authentication PAP | CHAP | MSCHAP	Specifies the Point-to-Point Protocol (PPP) authentication protocol. The Windows client dial-up networking settings lets you specify what authentication protocol to use (PAP, CHAP, or MS-CHAP). Whatever you specify on the client must match the setting you use on the PIX Firewall. Password Authentication Protocol (PAP) lets PPP peers authenticate each other. PAP passes the host name or username in clear text. Challenge Handshake Authentication Protocol (CHAP) lets PPP peers prevent unauthorized access through interaction with an access server. MS-CHAP is a Microsoft derivation of CHAP. PIX Firewall supports MS-CHAP Version 1 only (not Version 2.0). If an authentication protocol is not specified on the host, do not specify the **ppp authentication** option in your configuration.
ppp encryption mppe 40 | 128 | auto [**required**]	Specifies the number of session key bits used for MPPE (Microsoft Point-to-Point Encryption) negotiation. The domestic version of the Windows client can support 40- and 128-bit session keys, but international version of the Windows client only supports 40-bit session keys. On the PIX Firewall, use **auto** to accommodate both. Use **required** to indicate that MPPE must be negotiated or the connection will be terminated.
pppinterface id *intf_id*	A PPP virtual interface is created for each PPTP or PPPoE tunnel.
pptp echo *echo_timeout*	Specifies the PPTP keep-alive echo timeout value in seconds. PIX Firewall terminates a tunnel if an echo reply is not received within the timeout period you specify.
request **dialout pppoe**	Specifies to allow dialout PPPoE requests.
state	Session state.
store-local	Store in local Flash memory instead of using external configuration.
summary	Tunnel summary information.
transport	Tunnel transport information.
tunnel	[**clear** command only]—Removes one or more L2TP or PPTP tunnels from the configuration.

tunnel *tnl_id*	[**clear** command only]—Removes PPTP tunnels from the configuration that match *tnl_id*. You can view the tunnel IDs with the **show vpdn tunnel** command.
username *name*	Enter or display local username. However, when used as a **clear** command option, **username** removes all **vpdn username** commands from the configuration.
window	Window information.

Command Modes

Configuration mode.

Usage Guidelines

Virtual Private Dial-up Networking (VPDN) is used to provide long distance, point-to-point connections between remote dial-in users and a private network. VDPN uses Layer 2 tunnelling technologies (L2TP, PPTP, and PPPOE) to establish dial-up networking connections from the remote user to the private network across a public network.

Point-to-Point Tunneling Protocol (PPTP) is a Layer 2 protocol that tunnels the IP protocol. (For more details on PPTP, see RFC 2637, which describes the PPTP protocol.)

L2TP supports PPP by managing communications transactions. (There is a one-to-one relationship between a PPP connection and L2TP session.)

PPPOE is the Point-to-Point Protocol (PPP) over Ethernet. PPP is designed to work with network layer protocols such as IP, IPX, and ARA. PPP also has CHAP and PAP as built-in security mechanisms.

The **vpdn** command implements the L2TP, PPTP, and PPPoE features for the inbound connections. Refer to the *Cisco PIX Firewall and VPN Configuration Guide* for L2TP, PPTP, and PPPOE configuration examples.

Note The PIX Firewall is a PPTP and L2TP Server and a PPPoE client.

The **show vpdn tunnel** and **show vpdn session** commands display tunnel and session information (respectively) for LT2P (*l2tp*), PPTP (*pptp*), and PPPOE (**pppoe**). If you want to display information for only one protocol, use the option for that protocol. For example, the **show vpdn session pppoe** command displays session information for PPPOE sessions only.

The **clear vpdn** command removes all **vpdn** commands from the configuration and stops all the active PPTP, L2TP, and PPPoE tunnels. The **clear vpdn all** command lets you remove all tunnels, and the **clear vpdn id** *tnl_id* command lets you remove tunnels associated with *tnl_id*. (You can view the *tnl_id* with the **show vpdn** command.) The **clear vpdn group** command removes all the **vpdn group** commands from the configuration. The **clear vpdn username** command removes all the **vpdn username** commands from the configuration.

PPPoE

Because PPPoE encapsulates PPP, PPPoE relies on PPP to perform authentication and ECP and CCP functions for client sessions operating within the VPN tunnel. Additionally, PPPoE is not supported in conjunction with DHCP because PPP assigns the IP address for PPPoE.

The following are PPPoE restrictions on the PIX Firewall:

- The PIX Firewall acts as a PPPoE client only.
- The PPPoE client is only supported on the outside interface of the PIX Firewall in PIX Firewall software Version 6.2.

Note Unless the VPDN group for PPPoE is configured, PPPoE will not be able to establish a connection.

To define a VPDN group to be used for PPPoE, use the **vpdn group** *group_name* **request dialout pppoe** command.

If your ISP requires authentication, use the **vpdn group** *group_name* **ppp authentication PAP | CHAP | MSCHAP** command to select the authentication protocol used by your ISP.

Use the **vpdn group** *group_name* **localname** *username* command to associate the username assigned by your ISP with the VPDN group.

Use the **vpdn username** *username* password *pass* command to create a username and password pair for the PPPoE connection. The username must be a username that is already associated with the VPDN group specified for PPPoE.

Note If your ISP is using CHAP or MS-CHAP, the username may be called the remote system name and the password may be called the CHAP secret.

The PPPoE client functionality is turned off by default, so after VPDN configuration, enable PPPoE with the **ip address** *if_name* **pppoe** [**setroute**] command. The **setroute** option causes a default route to be created if no default route exists.

As soon as PPPoE is configured, the PIX Firewall attempts to find a PPPoE access concentrator with which to communicate. When a PPPoE connection is terminated, either normally or abnormally, the PIX Firewall attempts to find a new access concentrator with which to communicate.

The following **ip address** commands should not be used after a PPPoE session is initiated because they will terminate the PPPoE session:

- **ip address outside pppoe**, because it attempts to initiate a new PPPoE session.
- **ip address outside dhcp**, because it disables the interface until the interface gets its DHCP configuration.
- **ip address outside** *address netmask*, because it brings up the interface as a normally initialized interface.

PPTP

Use the **vpdn** command with the s**ysopt connection permit-pptp** to allow PPTP traffic to bypass checking of **conduit** or **access-list** command statements.

You can troubleshoot PPTP traffic with the **debug ppp** and **debug vpdn** commands.

PPTP is an alternative to IPSec handling for VPN clients or Easy VPN Remote devices. While PPTP is less secure than IPSec, PPTP is easier to implement and maintain. Only inbound PPTP connections are supported and only one PIX Firewall interface can have the **vpdn** command enabled.

Supported authentication protocols include: PAP, CHAP, and MS-CHAP using external AAA (RADIUS or TACACS+) servers or the PIX Firewall local username and password database. Through the PPP IPCP protocol negotiation, PIX Firewall assigns a dynamic internal IP address to the PPTP client allocated from a locally defined IP address pool.

PIX Firewall PPTP VPN supports standard PPP CCP negotiations with Microsoft Point-To-Point Encryption (MPPE) extensions using RSA/RC4 algorithm. MPPE currently supports 40-bit and 128-bit session keys. MPPE generates an initial key during user authentication and refreshes the key regularly. In this release, compression is not supported.

When you specify MPPE, you must use the MS-CHAP PPP authentication protocol. If you are using an external AAA server, the protocol must be RADIUS and the external RADIUS server must be able to return the Microsoft MSCHAP_MPPE_KEY attribute to the PIX Firewall in the RADIUS Authentication Accept packet. See RFC 2548, "Microsoft Vendor Specific RADIUS Attributes," for more information on the MSCHAP_MPPE_KEY attribute.

Cisco Secure ACS 2.5 and higher versions support the MSCHAP/MPPE encryption.

PIX Firewall PPTP VPN has been tested with the following Microsoft Windows products: Windows 95 with DUN 1.3, Windows 98, Windows NT 4.0 with Service Pack (SP) 6, and Windows 2000.

Note If you configure PIX Firewall for 128-bit encryption and if a Windows 95 or Windows 98 client does not support 128-bit or greater encryption, then the connection to the PIX Firewall is refused. When this occurs, the Windows client moves the dial-up connection menu down to the screen corner while the PPP negotiation is in progress. This gives the appearance that the connection is accepted when it is not. When the PPP negotiation completes, the tunnel terminates and PIX Firewall ends the connection. The Windows client eventually times out and disconnects.

Examples

The following is a sample PPPoE configuration:

```
vpdn group pppoegroup request dialout pppoe
vpdn group pppoegroup localname myusername
vpdn group pppoegroup ppp authentication pap
vpdn username myusername password mypassword

ip address outside pppoe setroute
```

The VPDN commands configure a VPDN group for PPPoE, and the **ip address outside pppoe setroute** command enables the PPPoE session.

The following is sample output from the **show vpdn tunnel l2tp** command:

```
pix# show vpdn tunnel l2tp

L2TP Tunnel Information (Total tunnels=1 sessions=1)

Tunnel id 1 is up, remote id is 7, 1 active sessions
 Tunnel state is established, time since change 12 secs
  Remote Internet Address 172.122.16.8, port 1701
  Local Internet Address 172.23.58.48, port 1701
  15 packets sent, 48 received, 377 bytes sent, 4368 received
  Control Ns 3, Nr 4
  Local RWS 16, Remote RWS 8
  Retransmission time 1, max 1 seconds
  Unsent queuesize 0, max 0
  Resend queuesize 0, max 1
  Total resends 0, ZLB ACKs 2
  Retransmit time distribution: 0 0 0 0 0 0 0 0 0
pix#
```

The following is sample output from the **show vpdn tunnel** command:

```
pix# show vpdn tunnel

L2TP Tunnel Information (Total tunnels=1 sessions=1)

Tunnel id 1 is up, remote id is 7, 1 active sessions
  Tunnel state is established, time since change 12 secs
 Remote Internet Address 172.122.16.8, port 1701
 Local Internet Address 172.23.58.48, port 1701
 15 packets sent, 48 received, 377 bytes sent, 4368 received
 Control Ns 3, Nr 4
 Local RWS 16, Remote RWS 8
 Retransmission time 1, max 1 seconds
 Unsent queuesize 0, max 0
 Resend queuesize 0, max 1
 Total resends 0, ZLB ACKs 2
 Retransmit time distribution: 0 0 0 0 0 0 0 0 0
% No active PPTP tunnels
pix#
```

The following is sample output from the **show vpdn tunnel packet** command:

```
show vpdn tunnel packet
PPTP Tunnel Information (Total tunnels=1 sessions=1)

LocID   Pkts-In  Pkts-Out  Bytes-In  Bytes-Out
    1      1196        13    113910        420
```

The following is sample output from the **show vpdn tunnel state** command:

```
show vpdn tunnel state
PPTP Tunnel Information (Total tunnels=1 sessions=1)


LocID RemID   State   Time-Since-Event-Chg
    1     1   estabd       6 secs
```

The following is sample output from the **show vpdn tunnel summary** command:

```
show vpdn tunnel summary
PPTP Tunnel Information (Total tunnels=1 sessions=1)

LocID RemID    State  Remote Address   Port  Sessions
    1     1   estabd   172.16.38.194   1723         1
```

The following is sample output from the **show vpdn tunnel transport** command:

```
show vpdn tunnel transport
PPTP Tunnel Information (Total tunnels=1 sessions=1)


LocID Type Local Address   Port  Remote Address  Port
    1  IP   172.16.1.209    1723  172.16.38.194   1723
```

The following is sample output from the **show vpdn session** command:

```
pix# show vpdn session
L2TP Session Information (Total tunnels=1 sessions=1)

Call id 1 is up on tunnel id 1
Remote tunnel name is abc-win2ke2
  Internet Address is 172.122.16.8
  Session username is guest, state is established
    Time since change 158 secs, interface outside
    Remote call id is 1
    PPP interface id is 1
    15 packets sent, 83 received, 377 bytes sent, 8412 received
      Sequencing is off

% No active PPTP tunnels
```

The following is sample output of a simple configuration that allows Windows PPTP clients to dial in without any authentication (not recommended). The Windows client can Telnet to internal host 192.168.0.2 through the static global address 209.165.201.2.

```
ip local pool my-addr-pool 10.1.1.1-10.1.1.254
vpdn group 1 accept dialin pptp
vpdn group 1 client configuration address local my-addr-pool
vpdn enable outside
static (inside, outside) 209.165.201.2 192.168.0.2
access-list acl_out permit tcp 10.1.1.0 255.255.255.0 host 209.165.201.2 eq telnet
access-group acl_out in interface outside
```

In the next example, PPTP clients authenticate using MS-CHAP and negotiate MPPE encryption with the PIX Firewall. The PPTP client can Telnet to host 192.168.0.2 through the static global 209.165.201.2. The Telnet session will be encrypted.

```
ip local pool my-addr-pool 10.1.1.1-10.1.1.254
aaa-server my-aaa-server-group (inside) host 192.168.0.10 key 12345678
aaa-server my-aaa-server-group protocol radius
vpdn group 1 accept dialin pptp
vpdn group 1 ppp authentication mschap
vpdn group 1 client authentication aaa my-aaa-server-group
vpdn group 1 ppp encryption mppe auto required
vpdn group 1 client configuration address local my-addr-pool
vpdn enable outside
static (inside, outside) 209.165.201.2 192.168.0.2
access-list acl_out permit tcp 10.1.1.0 255.255.255.0 host 209.165.201.2 eq telnet
access-group acl_out in interface outside
```

In the next example, PPTP clients authenticate using MS-CHAP, negotiate MPPE encryption, receive the DNS and WINS server addresses, and can Telnet to the host 192.168.0.2 directly through the **nat 0** command statement.

```
ip local pool my-addr-pool 10.1.1.1-10.1.1.254
aaa-server my-aaa-server-group (inside) host 192.168.0.10 key 12345678
aaa-server my-aaa-server-group protocol radius
vpdn group 1 accept dialin pptp
vpdn group 1 ppp authentication mschap
vpdn group 1 ppp encryption mppe auto required
vpdn group 1 client configuration address local my-addr-pool
vpdn group 1 client authentication aaa my-aaa-server-group
vpdn group 1 client configuration dns 10.2.2.99
vpdn group 1 client configuration wins 10.2.2.100
vpdn enable outside
access-list nonat permit ip host 192.168.0.2 10.1.1.0 255.255.255.0
access-list nonat permit ip host 10.2.2.99 10.1.1.0 255.255.255.0
access-list nonat permit ip host 10.2.2.100 10.1.1.0 255.255.255.0
nat (inside) 0 access-list nonat
access-list acl_out permit tcp 10.1.1.0 255.255.255.0 host 192.168.0.2 eq telnet
access-list acl_out permit udp 10.1.1.0 255.255.255.0 host 10.2.2.99 eq domain
access-list acl_out permit udp 10.1.1.0 255.255.255.0 host 10.2.2.100 eq netbios-ns
access-group acl_out in interface outside
```

In the next example, PPTP clients authenticate using MS-CHAP, negotiate MPPE encryption, receive the DNS and WINS server addresses, and can Telnet to the host 192.168.0.2 directly through the **nat 0** command statement. An **access-group** command statement is not present because the **sysopt connection permit-pptp** command statement allows all the PPTP traffic through the tunnel.

```
ip local pool my-addr-pool 10.1.1.1-10.1.1.254
aaa-server my-aaa-server-group (inside) host 192.168.0.10 key 12345678
aaa-server my-aaa-server-group protocol radius
vpdn group 1 accept dialin pptp
vpdn group 1 ppp authentication mschap
vpdn group 1 ppp encryption mppe auto required
vpdn group 1 client configuration address local my-addr-pool
vpdn group 1 client authentication aaa my-aaa-server-group
vpdn group 1 client configuration dns 10.2.2.99
vpdn group 1 client configuration wins 10.2.2.100
vpdn enable outside
access-list nonat permit ip host 192.168.0.2 10.1.1.0 255.255.255.0
access-list nonat permit ip host 10.2.2.99 10.1.1.0 255.255.255.0
access-list nonat permit ip host 10.2.2.100 10.1.1.0 255.255.255.0
nat (inside) 0 access-list nonat
sysopt connection permit-pptp
```

In the next example, PPTP clients authenticate using MS-CHAP, negotiate MPPE encryption, receive the DNS and WINS server addresses, and can Telnet to the host 192.168.0.2 directly through the **nat 0** command. The PPTP authenticates using the PIX Firewall local username and password database you create with the **vpdn username** command. Users are reauthenticated again by the **aaa** command when they start a Telnet session. An **access-group** command statement is not present because the **sysopt connection permit-pptp** command statement allows all the PPTP traffic through the tunnel.

```
ip local pool my-addr-pool 10.1.1.1-10.1.1.254
aaa-server my-aaa-server-group (inside) host 192.168.0.10 key 12345678
aaa-server my-aaa-server-group protocol radius
vpdn username usrname1 password password1
vpdn group 1 accept dialin pptp
vpdn group 1 ppp authentication mschap
vpdn group 1 ppp encryption mppe auto required
vpdn group 1 client configuration address local my-addr-pool
vpdn group 1 client authentication local
vpdn group 1 client configuration dns 10.2.2.99
vpdn group 1 client configuration wins 10.2.2.100
vpdn enable outside
access-list nonat permit ip host 192.168.0.2 10.1.1.0 255.255.255.0
access-list nonat permit ip host 10.2.2.99 10.1.1.0 255.255.255.0
access-list nonat permit ip host 10.2.2.100 10.1.1.0 255.255.255.0
nat (inside) 0 access-list nonat
sysopt connection permit-pptp
aaa authentication include telnet inbound 192.168.0.2 255.255.255.255 10.1.1.0
255.255.255.0
```

vpnclient

Configures Easy VPN Remote.

vpnclient vpngroup *group_name* **password** *preshared_key*

vpnclient username *xauth_username* **password** *xauth_password*

vpnclient server *ip_primary* [*ip_secondary_1 ip_secondary_2 ... ip_secondary_10*]

vpnclient mac-exempt *mac_addr_1 mac_mask_1* [*mac_addr_2 mac_mask_2*]

vpnclient mode client-mode | network-extension-mode

vpnclient management {[**tunnel** {*ip_addr_1 ip_mask_1*} [{i*p_addr_2 ip_mask_1*}...]] | [**clear**]}

no vpnclient management

[**no**] **vpnclient connect**

vpnclient disconnect

[**no**] **vpnclient nem-st-autoconnect**

vpnclient enable

no vpnclient {**server | mode | vpngroup | username | mac-exempt | management | enable**}

clear vpnclient

show vpnclient [**detail**]

Syntax Description

group_name	The name of the VPN group configured on the VPN headend. The maximum length is 63 characters and no spaces are permitted.
ip_addr_1, ip_addr_2, ...	The IP address of the remote network managing the client through the VPN tunnel.
ip_mask_1, ip_mask_2, ...	The IP mask of the remote network managing the client through the VPN tunnel.
ip_primary	The IP address of the primary Cisco Easy VPN Server.
ip_secondary_1, *ip_secondary_2*, ... , *ip_secondary_10*	The IP address of a secondary Cisco Easy VPN Server. There can be from 1 to 10 secondary Cisco Easy VPN Servers (backup VPN headends) configured. However, check your platform-specific documentation for applicable peer limits on your PIX Firewall platform.
mac_addr_n	The MAC address for user authentication exemption.
mac_mask_n	The MAC mask for user authentication exemption.
management clear	Specifies to use clear network traffic for management access to an Easy VPN Remote device.
management tunnel {*ip_addr_1 ip_mask_1*} [{i*p_addr_2 ip_mask_1*}...]	Specifies to use a VPN tunnel for management access to an Easy VPN Remote device.
nem-st-autoconnect	Specifies to automatically initiate IPSec data tunnels when split tunneling is configured. Note that IPSec data tunnels are automatically initiated and sustained when in network extension mode, except when split tunneling is configured.
password	Specifies to set the password.
preshared_key	The IKE pre-shared key used for authentication by the Easy VPN Server. The maximum length is 127 characters.
xauth_password	The user password to be used for XAUTH. The maximum length is 127 characters.
xauth_username	The username to be used for XAUTH. The maximum length is 127 characters.

Defaults

Easy VPN management is through the network by default.

Command Modes

Configuration mode.

Usage Guidelines

The **vpnclient** command stores non-transitory Easy VPN Remote device configuration information in the Flash memory of the PIX Firewall so that it is preserved whether or not the PIX Firewall reboots.

Note The PIX 501 and PIX 506/506E are both Easy VPN Remote and Easy VPN Server devices. The PIX 515/515E, PIX 525, and PIX 535 act as Easy VPN Servers only.

The PIX 501 and PIX 506/506E can act as Easy VPN Remote devices or Easy VPN Servers so that they can be used either as a client device or VPN headend in a remote office installation. The PIX 515/515E, PIX 525, and PIX 535 act as Easy VPN Servers only because the capacity of these devices makes them appropriate VPN headends for higher traffic environments.

Easy VPN management is through clear network traffic by default (**vpnclient management clear**). However, if Easy VPN management through a VPN tunnel is desired, use the **vpnclient management tunnel** {*ip_addr_1 ip_mask_1*} [{i*p_addr_2 ip_mask_1*}...] command.

You must specify all variables for the **vpnclient** configuration prior to enabling a Easy VPN Remote connection, except for the *xauth_username* and *xauth_password*. Also, you must configure NAT, IKE (using the **isakmp** and **isakmp policy** commands), the **crypto ipsec** transform set, **crypto map**, and an access control list (to trigger building the VPN tunnel) to enable Easy VPN Remote.

The **no vpnclient enable** command closes all established VPN tunnels and prevents new VPN tunnels from initiating until you enter a **vpnclient enable** command. The **no vpnclient connect** and **vpnclient disconnect** commands disconnect the existing VPN sessions but do not prevent new VPN tunnels from initiating.

The **clear vpnclient** command clears the Easy VPN Remote configuration and security policy stored in Flash memory.

The **show vpnclient** [**detail**] command displays VPN client or Easy VPN Remote device configuration information.The **show vpnclient** [**detail**] option displays dynamically generated configuration information.

vpnclient server

The **vpnclient server** *ip_primary ip_secondary_1*[*ip_secondary_2 ... ip_secondary_10*] command enables you to create a backup VPN server list on the VPN client.

If a backup server list is already configured locally on the VPN client, then it ignores any backup server configuration downloaded from the VPN headend.

If the VPN client has already downloaded a backup server configuration from the VPN headend and saved it to Flash memory, then you cannot configure a new backup server list locally until the headend deletes the downloaded list or you enter a **clear vpnclient** command on the VPN client.

Examples

The following is an example Easy VPN Remote configuration:

```
vpnclient vpngroup group_a password pre_share_a
vpnclient username user_1 password pass_1
vpnclient server 1.1.1.1
vpnclient mode client-mode
```

The following example sets up management access to an Easy VPN Remote device through a VPN tunnel:

```
vpnclient management tunnel 10.0.0.0 255.255.255.0
```

The following example sets up management access to an Easy VPN Remote device through clear network traffic:

```
vpnclient management clear
```

vpngroup

Supports Cisco VPN Client Version 3.x (Cisco Unified VPN Client Framework) and Easy VPN Remote devices.

vpngroup *group_name* **address-pool** *pool_name*

vpngroup *group_name* **authentication-server** *server_tag*

vpngroup *group_name* **backup-server** {{*ip1* [*ip2* **...** *ip10*]} | **clear-client-cfg**}

vpngroup *group_name* **default-domain** *domain_name*

vpngroup *group_name* **device-pass-through**

vpngroup *group_name* **dns-server** *dns_ip_prim* [*dns_ip_sec*]

vpngroup *group_name* **idle-time** *idle_seconds*

vpngroup *group_name* **max-time** *max_seconds*

vpngroup *group_name* **password** *preshared_key*

vpngroup *group_name* **pfs**

vpngroup *group_name* **secure-unit-authentication**

vpngroup *group_name* **split-dns** *domain_name1* [*domain_name2* **...** *domain_8*]

vpngroup *group_name* **split-tunnel** *access_list*

vpngroup *group_name* **user-authentication**

vpngroup *group_name* **user-idle-timeout** *user_idle_seconds*

vpngroup *group_name* **wins-server** *wins_ip_prim* [*wins_ip_sec*]

show vpngroup [*group_name*]

Syntax Description

access_list	The name of the access list for the split-tunnel configuration.
authentication-server *server_tag*	Specifies the IUA AAA server on the firewall headend.
backup-server	Configures a backup server list to be used for access by VPN clients if the primary server is not available.
clear-client-cfg	Clears backup servers from the client configuration.
device-pass-through	Specifies to exempt devices based on their MAC address from authentication. This may be used for devices such as Cisco IP Phones that cannot use IUA for authentication. Use with the **vpnclient mac-exempt** command.
dns_ip_prim	The IP address of the primary DNS server.
dns_ip_sec	The IP address of the secondary DNS server.
domain_name	The default domain name, up to 127 characters.

domain_name1 [*domain_name2*, *domain_name3*, ... , *domain_name8*]	The domains to configure for split DNS. The maximum length for a domain name is 127 characters.
group_name	Specifies the VPN policy group name and is an ASCII string with a maximum length of 63 characters. (You choose the name.)
idle_seconds	The idle timeout in seconds, from 60 to 86400. The default is 1800 seconds (30 minutes).
max_seconds	The maximum connection time in seconds that the VPN group is allowed, from 60 to 31536000. The default maximum connection time is set to unlimited.
pfs	Specifies to require that the VPN client or Easy VPN Remote device to perform PFS.
pool_name	The IP address pool name, up to 63 characters.
preshared_key	The VPN group pre-shared key. The maximum is 127 characters.
server_tag	AAA server tag to authenticate remote users of a hardware client.
split-dns	Specifies to use split DNS.
user_idle_seconds	Idle timeout for user authentication, in seconds.
vpngroup	Identifies the VPN dial-up group. The maximum identifier length is 63 characters.
wins_ip_prim	The IP address of the primary WINS server.
wins_ip_sec	The IP address of the secondary WINS server.

Command Modes

Configuration mode.

Usage Guidelines

Be sure to configure the IKE Mode Config prior to configuring support for the Cisco VPN 3000 Client. In configuring IKE Mode Config, specify that the PIX Firewall initiates the IKE Mode Config.

For additional information about configuring interoperability with the Cisco VPN 3000 Client using the **vpngroup** commands, see the *Cisco PIX Firewall and VPN Configuration Guide*.

The Cisco VPN 3000 Client supports Windows 2000.

The **vpngroup** command set lets you configure Cisco VPN 3000 Client policy attributes to be associated with a VPN group name and downloaded to the Cisco VPN 3000 Client(s) that are part of the given group. The same VPN group name is configured in the Cisco VPN 3000 Client to ensure the matching of VPN client or Easy VPN Remote policy.

Configure a VPN group name of “default” to create a VPN group policy that matches any group name. The PIX Firewall selects the VPN group name “default,” if there is no other policy match.

The **vpngroup address-pool** command lets you define a pool of local addresses to be assigned to a VPN group.

Note Both the **vpngroup address-pool** command and the **ip local pool** command enable you to specify a pool of local addresses to be used for assigning dynamic IP addresses to VPN clients and Easy VPN Remote devices. In the case of the Cisco VPN 3000 Client, the specified pool of addresses is associated with a given group, which consists of Cisco VPN 3000 Client users. We recommend using the **vpngroup address-pool** command only if you will configure more than one pool of addresses to be used by more than one VPN user group. The **vpngroup address-pool** command gives the PIX Firewall added flexibility to configure different pools of local addresses for different user groups.

Individual User Authentication (IUA) is a centrally managed feature that cannot be configured locally, but it must be enabled locally. The **vpngroup** *group_name* **user-authentication** command enables IUA on the firewall.

The **vpngroup** *group_name* **secure-unit-authentication** command enables Secure Unit Authentication (SUA) for the vpngroup. SUA is a centrally managed feature and cannot be directly configured on Easy VPN Remote devices. If SUA is enabled, a downloaded VPN policy activates SUA on the Easy VPN Remote device. SUA can be disabled by a corresponding VPN policy. SUA status reverts to UNSPECIFIED if a **clear vpnclient** command is entered on the firewall.

The **vpngroup** *group_name* **user-idle-timeout** *user_idle_seconds* command sets the IUA idle timeout.

The **vpngroup dns-server** command enables the PIX Firewall to download an IP address of a DNS server to a Cisco VPN 3000 Client as part of an IKE negotiation.

The **vpngroup wins-server** command lets the PIX Firewall download an IP address of a WINS server to a Cisco VPN 3000 Client as part of an IKE negotiation.

To enable the PIX Firewall to download a default domain name to a Cisco VPN 3000 Client as part of IKE negotiation, use the **vpngroup default-domain** command.

Use the **vpngroup split-tunnel** command to enable split tunneling on the PIX Firewall. Split tunneling allows a remote VPN client or Easy VPN Remote device simultaneous encrypted access to the corporate network and clear access to the Internet. Using the **vpngroup split-tunnel** command, specify the access list name to which to associate the split tunnelling of traffic. With split tunnelling enabled, the PIX Firewall downloads its local network IP address and netmask specified within the associated access list to the VPN client or Easy VPN Remote device as part of the policy push to the client. In turn, the VPN client or Easy VPN Remote device sends the traffic destined to the specified local PIX Firewall network via an IPSec tunnel and all other traffic in the clear. The PIX Firewall receives the IPSec-protected packet on its outside interface, decrypts it, and then sends it to its specified local network.

If you do not enable split tunneling, all traffic between the VPN client or Easy VPN Remote device and the PIX Firewall is sent through an IPSec tunnel. All traffic originating from the VPN client or Easy VPN Remote device is sent to the PIX Firewall's outside interface through a tunnel, and the client's access to the Internet from its remote site is denied.

Regardless of whether split tunneling is enabled, VPN clients and Easy VPN Remote devices negotiate an IPSec tunnel to the PIX Firewall unit's IP address with a netmask of 255.255.255.255.

Networks defined in **access-list deny** command statements are not pushed to VPN clients or Easy VPN Remote devices.

The **vpngroup idle-time** command sets the inactivity timeout for a Cisco VPN 3000 Client. When the inactivity timeout for all IPSec SAs have expired for a given VPN client or Easy VPN Remote device, the tunnel is terminated. The default inactivity timeout is 30 minutes.

The **vpngroup max-time** command sets the maximum connection time for a Cisco VPN 3000 Client. When the maximum connection time is reached for a given VPN client or Easy VPN Remote device, the tunnel is terminated. This means the connection between the Cisco VPN 3000 Client and the PIX Firewall will have to be reestablished. The default maximum connection time is set to an unlimited amount of time.

Note The inactivity timeout specified with the **vpngroup idle-time** command and maximum connection time specified with the **vpngroup max-time** command for a given Cisco VPN 3000 Client take precedence over the commands used to set global lifetime timeouts. These commands are the **isakmp policy lifetime and crypto map set security-association lifetime seconds** commands.

Configure the VPN group's pre-shared key employing the **vpngroup password** command to be used during IKE authentication. This pre-shared key is equivalent to the password that you enter within the **Group Password** box of the Cisco VPN 3000 Client while configuring your group access information for a connection entry.

The PIX Firewall configured password displays in asterisks within the file configuration.

Note Both the **vpngroup password** command and the **isakmp key address** command let you specify a pre-shared key to be used for IKE authentication. We recommend that you use the **vpngroup password** command only if you plan to configure more than one VPN user group. The **vpngroup password** command gives the PIX Firewall added flexibility to configure different VPN user groups.

Examples

The following example show use of the **vpngroup** commands. The VPN client(s) or Easy VPN Remote device(s) within the VPN group named as "myVpnGroup" will be dynamically assigned one of the IP addresses from the pool of addresses ranging from 10.140.40.0 to 10.140.40.7. The policy attributes for the group "myVpnGroup" will be downloaded to the given VPN client or Easy VPN Remote device during the policy push to the client. Split tunnelling is enabled. In the example, all traffic destined for the 10.130.38.0 255.255.255.0 PIX Firewall network from the VPN client or Easy VPN Remote device will be IPSec protected.

```
access-list 90 permit ip 10.130.38.0 255.255.255.0 10.140.40.0 255.255.255.248

ip local pool vpnpool 10.140.40.1-10.140.40.7

crypto ipsec transform-set esp-sha esp-null esp-sha-hmac
crypto dynamic-map dynmap 50 set transform-set esp-sha
crypto map mapName 10 ipsec-isakmp dynamic dynmap
crypto map mapName client configuration address initiate
crypto map mapName interface outside

isakmp enable outside
isakmp identity hostname
isakmp policy 7 authentication pre-share
isakmp policy 7 encryption 3des
isakmp policy 7 hash md5
isakmp policy 7 group 1

vpngroup myVpnGroup address-pool vpnpool
vpngroup myVpnGroup dns-server 10.131.31.11
vpngroup myVpnGroup wins-server 10.131.31.11
vpngroup myVpnGroup default-domain example.com
vpngroup myVpnGroup split-tunnel 90
vpngroup myVpnGroup idle-time 1800
```

```
vpngroup myVpnGroup max-time 86400
vpngroup myVpnGroup password ********
```

who

Show active Telnet administration sessions on the PIX Firewall.

who [*local_ip*]

show who [*local_ip*]

Syntax Description

local_ip	An optional internal IP address to limit the listing to one IP address or to a network IP address.

Command Modes

Unprivileged mode.

Usage Guidelines

The **who** command shows the PIX Firewall TTY_ID and IP address of each Telnet client currently logged into the PIX Firewall. This command is the same as the **show who** command.

Examples

The following example shows how to display the current Telnet sessions:

```
pixfirewall# who

0: From 192.168.1.3
1: From 192.168.2.2
```

Related Commands

- **kill**
- **telnet**

write

Store, view, or erase the current configuration.

write net [[*server_ip*]**:**[*filename*]]

write erase

write floppy

write memory | floppy [**uncompressed**]

write standby

write terminal

Note The PIX 506/506E does not support use of the **write standby** command. Also, the PIX 506/506E, PIX 515/515E, and the PIX 525 do not support use of the **write floppy** command.

Syntax Description

erase	Clear the Flash memory configuration.
filename	A filename you specify to qualify the location of the configuration file on the TFTP server named in *server_ip*. If you set a filename with the **tftp-server** command, do not specify it in the **write** command; instead just use a colon (:) without a filename. Many TFTP servers require the configuration file to be world-writable to write to it.
floppy	Stores the current configuration on diskette.
memory	Stores the current configuration in Flash memory, along with the activation key value and timestamp for when the configuration was last modified.
server_ip	Specifes the IP address of the TFTP server. If you specify the full path and filename in the **tftp-server** command, then use a ":" in the **write** command.
standby	Stores the configuration to the failover standby unit from RAM-to-RAM.
terminal	Display current configuration on the terminal.
uncompressed	Writes the configuration to memory without storing it in compressed format.

Command Modes

Privileged mode.

Usage Guidelines

The **write net** command stores the current configuration into a file on a TFTP server elsewhere in the network. Additionally, the **write net** command uses the TFTP server IP address specified in the **tftp-server** command. If you specify both the IP address and path name in the **tftp-server** command, you can specify the **write net :***filename* command as simply a colon (:) as follows:

```
write net :
```

Use the **configure net** command to get the configuration from the file.

The **write erase** command clears the Flash memory configuration.

The **write floppy** command stores the current configuration on diskette. The diskette must be DOS formatted or a PIX Firewall boot disk. If you are formatting the diskette from Windows, choose the Full format type, not the Quick (erase) selection. You can tell that information is stored on the diskette by observing that the light next to the diskette drive glows while information transfers.

The diskette you create can only be read or written by the PIX Firewall. If you use the **write floppy** command with a diskette that is not a PIX Firewall boot disk, do not leave the floppy in the floppy drive because it will prevent the firewall from rebooting in the event of a power failure or system reload. Only one copy of the configuration can be stored on a single diskette.

The **write memory** command saves the current running configuration to Flash memory. Use the **configure memory** command to merge the current configuration with the image you saved in Flash memory.

PIX Firewall lets processing continue during the **write memory** command.

If another PIX Firewall console user tries to change the configuration while you are executing the **write memory** command, the user receives the following messages:

```
Another session is busy writing configuration to memory
Please wait a moment for it to finish
```

After the **write memory** command completes, PIX Firewall lets the other command complete.

Note Only use the **write memory** command if a configuration has been created with IP addresses for both network interfaces.

The **write standby** command writes the configuration stored in RAM on the active failover unit to the RAM on the standby unit. When the primary unit boots it automatically writes the configuration to the secondary unit. Use the **write standby** command if the primary and secondary units' configurations have different information.

The **write terminal** command displays the current configuration in the PIX Firewall unit's RAM memory.

You can also display the configuration stored in Flash memory using the **show configure** command.

Defaults

The default on the PIX Firewall is to store all configurations in compressed format. However, whether a configuration is stored compressed or uncompressed is transparent when executing configuration commands.

Examples

The following example specifies the TFTP server and creates a file named **new_config** in which to store the configuration:

```
tftp-server 10.1.1.2 /pixfirewall/config/new_config
write net :
```

The following example erases the contents of Flash memory and reloads the PIX Firewall:

```
write erase
Erase PIX configuration in Flash memory? [confirm] y
reload
Proceed with reload? [confirm] y
```

The following example saves the configuration on diskette:

```
write floppy
Building configuration…
[OK]
```

The following example saves the current configuration to Flash memory:

```
write memory
Building configuration…
[OK]
```

The following example displays the configuration:

```
write terminal
Building configuration…
: Saved
…
```

Related Commands

- configure

Y and Z Commands

There are no “y” or “z” PIX Firewall commands.

Numerics

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X