



T through X Commands

telnet

Specify host for PIX Firewall console access via Telnet. (Configuration mode.)

```
telnet ip_address [netmask] [if_name]  
clear telnet [ip_address [netmask] [if_name]]  
no telnet [ip_address [netmask] [if_name]]  
show telnet  
telnet timeout minutes  
show telnet timeout
```

Syntax Description

<i>ip_address</i>	An IP address of a host or network that can access the PIX Firewall Telnet console. 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.
<i>netmask</i>	Bit mask of <i>ip_address</i> . To limit access to a single IP address, use 255 in each octet; for example, 255.255.255.255. If you do not specify <i>netmask</i> , it defaults to 255.255.255.255 regardless of the class of <i>local_ip</i> . Do not use the subnetwork mask of the internal network. The <i>netmask</i> is only a bit mask for the IP address in <i>ip_address</i> .
<i>if_name</i>	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.
timeout <i>minutes</i>	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.

Usage Guidelines

The **telnet** command allows you to 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.

Up to 16 hosts or networks are allowed access to the PIX Firewall console with Telnet, 5 simultaneously. The **show telnet** command displays the current list of IP addresses authorized to access 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 console session.

If the **aaa** command is used with the **console** option, Telnet console access must be authenticated with an authentication server.

**Note**

If you have configured the **aaa** command to require authentication for PIX Firewall Telnet console 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 console 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 console from any of these internal interfaces. An additional facet of this behavior is that you have to 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 Chapter 9, "Upgrading PIX Firewall Software" 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.

Related Commands

- [aaa](#)
- [kill](#)
- [passwd](#)
- [who](#)

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 console 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>
```

terminal

Change console terminal settings. (Configuration mode.)

terminal [no] monitor

terminal width *characters*

Syntax Description

monitor	Enable or disable syslog message displays on the console.
width	Set the width for displaying information during console sessions.
<i>characters</i>	Permissible values are 0, which means 511 characters, or a value in the range of 40 to 511.

Usage Guidelines

The **terminal monitor** command allows you to 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. (Configuration mode.)

```
tftp-server [if_name] ip_address path
```

```
no tftp-server [[if_name] ip_address path]
```

```
clear tftp-server [[if_name] ip_address path]
```

```
show tftp-server
```

Syntax Description

<i>if_name</i>	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.
<i>ip_address</i>	The IP address or network of the TFTP server.
<i>path</i>	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.

Usage Guidelines

The **tftp-server** command allows you to 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.

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. (Configuration mode.)

```
timeout [xlate [hh:mm:ss]] [conn [hh:mm:ss]] [half-closed [hh:mm:ss]] [udp [hh:mm:ss]]
[rpc [hh:mm:ss]] [h323 [hh:mm:ss]] [sip [hh:mm:ss]] [sip_media [hh:mm:ss]][uauth
[hh:mm:ss] [absolute | inactivity]]
```

```
clear timeout
```

```
show timeout
```

Syntax Description

xlate <i>hh:mm:ss</i>	Idle time until a translation slot is freed. This duration must be at least 1 minute. The default is 3 hours.
conn <i>hh:mm:ss</i>	Idle time until a connection slot is freed. 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.
half-closed <i>hh:mm:ss</i>	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.
udp <i>hh:mm:ss</i>	Idle time until a UDP slot is freed. This duration must be at least 1 minute. The default is 2 minutes.
rpc <i>hh:mm:ss</i>	Idle time until an RPC slot is freed. This duration must be at least 1 minute. The default is 10 minutes.
sip <i>hh:mm:ss</i>	Modifies the SIP timer. SIP signalling port is set to a default of 30 minutes.
sip_media <i>hh:mm:ss</i>	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.
h323 <i>hh:mm:ss</i>	Duration for H.323 inactivity timer. When this time elapses, the port used by the H.323 service closes. This duration must be at least 5 minutes. The default is 5 minutes.
uauth <i>hh:mm:ss</i>	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.
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).
inactivity	Start uauth timer after a connection becomes idle.

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.

**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.

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 reprompted 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.

Use the **show timeout** command to display the current **timeout** command settings.

See also: **show xlate**, **uauth**.

**Note**

RPC and NFS are very insecure 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
```

uauth (clear and show)

Delete all authorization caches for a user. (Privileged mode.)

```
clear uauth [username]
```

```
show uauth [username]
```

Syntax Description

<i>username</i>	Clear or view user authentication information by username.
-----------------	--

Usage Guidelines

The **clear uauth** command deletes one user's or all users' AAA authorization caches, which forces the user or users to reauthenticate the next time they create a connection. 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 **show uauth** command also lists CiscoSecure 2.1 and later idletime and timeout values, which can be set for different user groups.

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 unproxies 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.

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.

Related Commands

- [aaa authorization](#)
- [timeout](#)

Examples

The following is sample output from the **show uauth** command:

```
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
```

url-cache

Cache responses to URL filtering requests to the Websense server. (Configuration mode.)

url-cache dst | src_dst size

no url-cache dst | src_dst size

clear url-cache

show url-cache stat

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 Websense server.
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 Websense server.
<i>size</i>	Specify a value for the cache size within the range 1 to 128 KB.
stat	Use the stat option to display additional URL cache statistics, including the number of cache lookups and hit rate.

Usage Guidelines

The **url-cache** command caches responses to URL filtering requests to the Websense server. 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 Websense server. Disable caching with the **no url-cache** command. The **clear url-cache** command removes **url-cache** command statements from the configuration.

Access to the URL cache does not update the Websense accounting logs. Before using this command, let Websense run to accumulate logs to let you view Websense accounting information. After you get a usage profile that meets your security needs, enable this command to increase throughput.



Note

If you change settings on the Websense server, disable the cache with the **no url-cache** command and then re-enable the cache with the **url-cache** command.

The **url-cache** command allows you to enable URL caching, set the size of the cache, and displays cache statistics.

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 Websense access 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 Websense for use with the **filter** command. (Configuration mode.)

```
url-server [(if_name)] host ip_address [timeout seconds] [protocol [TCP | UDP] version [1 | 4]
```

```
no url-server host ip_address
```

Syntax Description

if_name	The network interface where the authentication server resides. If not specified, the default is inside.
host ip_address	The server that runs the Websense 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.
version	The version of the protocol can be configured using 1 or 4 keywords. 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.

Usage Guidelines

The **url-server** command designates a server that runs Websense, a URL filtering application. Once you designate the server, enable the URL filtering service with the **filter** command.



Note

You can have a total of 16 URL servers.

Follow these steps to filter URLs:

- Step 1** Designate a Websense server with 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 pdm** commands to view run information.

Additional 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 http 0 0 0 0
filter url except 10.0.2.54 255.255.255.255 0 0
```

virtual

Access PIX Firewall virtual server. (Configuration mode.)

virtual http *ip_address* [**warn**]

virtual telnet *ip_address*

Syntax Description

<i>ip_address</i>	<p>For outbound use, <i>ip_address</i> must be an address routed to the PIX Firewall. Use an RFC 1918 address that is not in use on any interface.</p> <p>For inbound use, <i>ip_address</i> must be an unused global address. An access-list and static command pair must provide access to <i>ip_address</i>, as well as an aaa authentication command statement. See the “Examples” section for more information.</p> <p>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 <i>ip_address</i> 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 <i>ip_address</i> 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 <i>ip_address</i>, then on the client, setting the PIX Firewall at 192.168.0.1 as the route to host 10.0.0.1.</p>
warn	<p>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.</p>

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 global IP address in the **static** command must be a real IP address. The local address in the **static** command is the IP address of the virtual server.

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) 209.165.201.1 192.168.1.1 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 192.168.1.1 255.255.255.255 0 0 tacacs+
virtual http 209.165.201.1
```

The next example displays the **show virtual** command output:

```
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 10.8.8.11 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
```

On an inside host:

```
/unix/host%telnet 209.165.201.30
Trying 209.165.201.30...
Connected to 209.165.201.30.
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.

vpng

Implement the L2TP or PPTP feature. (Configuration mode.)

```

vpng enable if_name

vpng group name accept dialin pptp|l2tp

vpng group name l2tp tunnel hello <hello_timeout>

vpng group group_name ppp authentication PAP | CHAP | MSCHAP

vpng group group_name ppp encryption mppe 40 | 128 | auto [required]

vpng group group_name client configuration address local address_pool_name

vpng group group_name client configuration dns dns_server_ip1 [dns_server_ip2]

vpng group group_name client configuration wins wins_server_ip1 [wins_server_ip2]

vpng group group_name client authentication aaa aaa_server_group

vpng group group_name client authentication local

vpng group group_name client accounting aaa_server_group

vpng username username password password

vpng group group_name pptp echo echo_timeout

show vpng tunnel [l2tp | pptp] [id tunnel_id | packets | state | summary | transport]

show vpng username [username]

show vpng session [l2tp | pptp] [id session_id | packets | state | window]

show vpng pppinterface [id intf_id]

clear vpng [group | username | tunnel [all | [id tunnel_id]]]

```

Syntax Description

enable <i>if_name</i>	Enable the VPDN function on a PIX Firewall interface. Specify the interface in <i>if_name</i> where L2TP or PPTP traffic is received. Only inbound connections are supported.
group <i>group_name</i>	Specify the VPDN group name. The VPDN <i>group_name</i> is an ASCII string to denote a VPDN group. You can make up the name. The maximum length of the name is 128 bytes.
accept dialin pptp l2tp pptp	Accept a dial-in request using PPTP or L2TP.

ppp authentication PAP CHAP MSCHAP	Specify the Point-to-Point Protocol (PPP) authentication protocol. The Windows client dial-up networking settings allows you to 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]	Specify 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.
client configuration address local <i>address_pool_name</i>	Specify 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 <i>dns_server_ip1</i> [<i>dns_server_ip2</i>]	Specify 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 <i>wins_server_ip1</i> [<i>wins_server_ip2</i>]	Specify up to two WINS server IP addresses.
client authentication aaa <i>aaa_server_group</i>	Specify 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 accounting aaa-server-group	Specify the AAA server group for accounting. The accounting aaa server group can be different from the aaa server group for user authentication.
password	Specify local user password.
pptp echo <i>echo_timeout</i>	Specify 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.
l2tp tunnel hello < <i>hello_timeout</i> >	Specify L2TP tunnel keep-alive hello timeout value in seconds. Default is 60 seconds if not specified. The value can be between 10 to 300 seconds.
show vpdn tunnel	Display tunnel information.
show vpdn session	Display session information.
l2tp pptp	Select either l2tp or pptp to display that tunnel information. The PIX Firewall shows both tunnel protocols if this option is not specified.
id	Identify tunnel or session.
id <i>tunnel_id</i>	Unique tunnel identifier.
id <i>session_id</i>	Unique session identifier.

pppinterface id <i>intf_id</i>	A PPP virtual interface is created for each PPTP tunnel. Use the show vpng session command to display the interface identification value.
username	Enter or display local username.
packets	Packet and byte count.
state	Session state.
summary	Tunnel summary information.
transport	Tunnel transport information.
window	Window information.
group	[clear command only]—Removes all vpng group commands from the configuration.
username	[clear command only]—Removes all vpng username commands from the configuration.
tunnel	[clear command only]—Removes one or more L2TP or PPTP tunnels from the configuration.
all	[clear command only]—Removes all L2TP or PPTP tunnels from the configuration.
id <i>tunnel_id</i>	[clear command only]—Removes PPTP tunnels from the configuration that match <i>tunnel_id</i> . You can view the tunnel IDs with the show vpng tunnel command.

Usage Guidelines

The **vpng** command implements the L2TP and PPTP feature for the inbound connection. Refer to *Cisco PIX Firewall and VPN Configuration Guide* for the L2TP configuration example. Point-to-Point Tunneling Protocol (PPTP) is a layer 2 tunneling protocol, which lets a remote client use a public IP network to communicate securely with servers at a private corporate network. PPTP tunnels the IP protocol. RFC 2637 describes the PPTP protocol.

Only inbound PPTP connections are supported and only one PIX Firewall interface can have the **vpng** command enabled.

PPTP is an alternative to IPSec handling for VPN clients. While PPTP is less secure than IPSec, PPTP is easier to implement and maintain.

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 later release 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.

You can troubleshoot PPTP traffic with the **debug ppp** and **debug vpdn** commands.

Use the **vpdn** command with the **sysopt connection permit-pptp** to allow PPTP traffic to bypass checking of **conduit** or **access-list** command statements.

The **show vpdn** commands list tunnel and session information.

The **clear vpdn** command removes all **vpdn** commands from the configurations and stops all the active PPTP tunnels. The **clear vpdn all** command allows you to remove all tunnels, and the **clear vpdn id tunnel_id** command allows you to remove tunnels associated with *tunnel_id*. (You can view the *tunnel_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. The **clear vpdn** command removes all **vpdn** commands from the configuration.

Examples

The following examples list the output from the **show vpdn** commands.

The following example 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 171.69.39.85, 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
pix#
```

The following example 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 171.69.39.85, 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 vpng tunnel packet** command:

```

show vpng 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 vpng tunnel state** command:

```

show vpng tunnel state
PPTP Tunnel Information (Total tunnels=1 sessions=1)

LocID RemID   State   Time-Since-Event-Chg
   1     1   estab   6 secs

```

The following is sample output from the **show vpng tunnel summary** command:

```

show vpng tunnel summary
PPTP Tunnel Information (Total tunnels=1 sessions=1)

LocID RemID   State  Remote Address  Port  Sessions
   1     1   estab  172.16.38.194  1723  1

```

The following is sample output from the **show vpng tunnel transport** command:

```

show vpng 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 vpng session** command:

```

pix# show vpng 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 171.69.39.85
  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 username1 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
```

vpngroup

New functionality has been added to implement the Point-to-Point Tunneling Protocol (PPTP) and Layer 2 Tunneling Protocol (L2TP) feature within virtual private dial-up network (VPDN) groups.

```
vpdn group group_name accept dialin [pptp | l2tp]
vpdn group group_name l2tp tunnel hello [hello_timeout]
vpdn group group_name client accounting [aaa_server_tag]
```

Implements support for the Cisco VPN 3000 Client. (Configuration mode.)

```
vpngroup group_name address-pool ip pool name
no vpngroup group_name address-pool ip pool name
vpngroup group_name default-domain domain_name
no vpngroup group_name default-domain domain_name
vpngroup group_name dns-server dns_ip_prim [dns_ip_sec]
no vpngroup group_name dns-server dns_ip_prim [dns_ip_sec]
vpngroup group_name idle-time idle_seconds
no vpngroup group_name idle-time idle_seconds
vpngroup group_name max-time max_seconds
no vpngroup group_name max-time max_seconds
vpngroup group_name password preshared_key
no vpngroup group_name password preshared_key
vpngroup group_name split-tunnel acl_name
no vpngroup group_name split-tunnel acl_name
vpngroup group_name wins-server wins_ip_prim [wins_ip_sec]
no vpngroup group_name wins-server wins_ip_prim [wins_ip_sec]
```

Syntax Description

<i>accept dialin</i>	Accept PPTP or L2TP dial-in request.
<i>group_name</i>	Specify the VPDN group name. The VPDN <i>group_name</i> is an ASCII string to denote a VPDN group. You can make up the name. The maximum length of the name is 128 bytes.
<i>pool_name</i>	The IP address pool name.
<i>dns_ip_prim</i>	The IP address of the primary DNS server.
<i>dns_ip_sec</i>	The IP address of the secondary DNS server.
<i>wins_ip_prim</i>	The IP address of the primary WINS server.
<i>wins_ip_sec</i>	The IP address of the secondary WINS server.
<i>domain_name</i>	The default domain name.

<i>acl_name</i>	The name of the access-list to which to bind split-tunneling.
<i>idle_seconds</i>	The inactivity timeout in seconds. The default is 1800 seconds or 30 minutes.
<i>max_seconds</i>	The maximum connection time in seconds the VPN group is allowed. The default maximum connection time is set to unlimited.
<i>preshared_key</i>	The VPN group pre-shared key.
vpngroup	Identify the virtual private dial-up network group.
pptp l2tp	Select PPTP or L2TP protocol.
l2tp tunnel hello	Specify the L2TP keep-alive hello timeout value. The default is 60 seconds if not specified. The minimum is 10 seconds and maximum is 300 seconds.
<i>hello_timeout</i>	Tunnel hello keep-alive message timeout period (in seconds).
client accounting	Generate AAA accounting start and stop record for the L2TP (and PPTP) session.
<i>aaa_server_tag</i>	The <i>aaa_server_tag</i> defined from the aaa-server command. The AAA server does not need to be the same server as the AAA authentication server.

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 **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 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 remote VPN clients. 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.

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 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 as part of the policy push to the client. In turn, the VPN client 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 and the PIX Firewall is sent through an IPSec tunnel. All traffic originating from the VPN client 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, the VPN client negotiates 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 the VPN client.

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, 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, 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** field 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) 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 a given VPN client 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 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 *****

```

The following examples show different configurations of the **vpdn group** command with L2TP.

```

vpdn group 1 accept dialin l2tp
vpdn group 1 l2tp tunnel hello 60
vpdn group 1 client accounting myaaa

```

who

Show active Telnet administration sessions on the PIX Firewall. (Unprivileged mode.)

```
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.

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.

See also: **kill**, **telnet**.

Examples

The following example shows how to display the current Telnet sessions:

```
who  
2: From 192.168.2.2  
1: From 192.168.1.3
```

write

Store, view, or erase the current configuration. (Privileged mode.)



Note

The PIX 506 does not support use of the **write standby** command. Also, the PIX 515, PIX 506, and the PIX 525 do not support use of the **write floppy** command.

write net *[[server_ip]:[filename]]*

write erase

write floppy

write memory

write standby

write terminal

Syntax Description

<i>server_ip</i>	Store current configuration at a host available across the network. If you specify the full path and filename in the tftp-server command, only specify a colon (:) in the write command.
<i>filename</i>	A filename you specify to qualify the location of the configuration file on the TFTP server named in <i>server_ip</i> . 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.
erase	Clear the Flash memory configuration.
floppy	Store current configuration on diskette.
memory	Store current configuration in Flash memory.
standby	Store configuration to the failover standby unit from RAM to RAM.
terminal	Display current configuration on the terminal.

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* as simply a colon (:):

```
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.

Examples

The following example specifies a configuration file on the TFTP server and then stores the configuration in the new_config file:

```
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
```

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](#)

xlate (clear and show)

View or clear translation slot information. (Privileged mode.)

```
clear xlate [global | local ip1[-ip2] [netmask mask]] lport | gport port[-port]
           [interface if1[,if2][,ifn]] [state static [,dump] [,portmap] [,norandomseq] [,identity]]
```

```
show xlate [global | local ip1[-ip2] [netmask mask]] lport | gport port[-port]
           [interface if1[,if2][,ifn]] [state static [,dump] [,portmap] [,norandomseq] [,identity]]
           [debug] [count]
```

Syntax Description

[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.
lport gport port[-port]	Display active translations by local and global port specifications. See “Ports” in Chapter 1, “Using PIX Firewall Commands” for a list of valid port literal names.
interface if1[,if2][,ifn]	Display active translations by interface.
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).

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.

Examples

The following is sample output for 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
```

Related Commands

- [show conn](#)
- [timeout](#)
- [uauth \(clear and show\)](#)

■ xlate (clear and show)