



## C Commands

---

### ca

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

```
ca authenticate ca_nickname [fingerprint]
```

```
ca configure ca_nickname ca | ra retry_period retry_count [crloptional]
```

```
no ca configure ca_nickname
```

```
show ca configure
```

```
ca crl request ca_nickname
```

```
no ca crl
```

```
ca enroll ca_nickname challenge_password [serial] [ipaddress]
```

```
no ca enroll ca_nickname
```

```
ca generate rsa key | specialkey key_modulus_size
```

```
ca identity ca_nickname ca_ipaddress[:ca_script_location] [ldap_ip address]
```

```
no ca identity ca_nickname
```

```
show ca identity
```

```
ca save all
```

**no ca save all**

**ca zeroize rsa** [*keypair\_name*]

**show ca certificate**

**show ca crl**

**show ca mypubkey rsa**

Syntax Description		
<i>ca_nickname</i>		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.
<i>fingerprint</i>		A key consisting of alphanumeric characters the PIX Firewall uses to authenticate CA's certificate.
<b>ca   ra</b>		Indicates whether to contact the CA or Registration Authority (RA) when using the <b>ca configure</b> command.  Some CA systems provide an RA, which the PIX Firewall contacts instead of the CA.
<i>retry_period</i>		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.
<i>retry_count</i>		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.
<b>crloptional</b>		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 <b>crloptional</b> option.
<i>challenge_password</i>		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.
<b>serial</b>		Return the PIX Firewall unit's serial number in the certificate.
<b>ipaddress</b>		Return the PIX Firewall unit's IP address in the certificate.
<b>key</b>		This specifies that one general-purpose RSA key pair will be generated.
<b>specialkey</b>		This specifies that two special-purpose RSA key pairs will be generated instead of one general-purpose key.
<i>key_modulus_size</i>		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.
<i>ca_ipaddress</i>		The CA's IP address.

<code>:ca_script_location</code>	The default location and script on the CA server is <code>/cgi-bin/pkiclient.exe</code> . If the CA administrator has not put the CGI script in this location, provide the location and the name of the script in the <b>ca identity</b> 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.
<code>ldap_ipaddress</code>	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.

### 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. See the Chapter 4, "Basic VPN Configuration" in the *Cisco PIX Firewall and VPN Configuration Guide* for a list of specific CA server versions the PIX Firewall supports.



#### Note

If you are using the VeriSign CA, you must use the **crloptional** parameter with the **ca configure** command.

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

#### 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 the **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.

## Examples

In this 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 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.



#### Note

---

When using PIX Version 6.0 or higher, it is no longer necessary to always use the **crloptional** option with the **ca configure** command. VeriSign as your CA.

---

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 replacing 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
```

### 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 need to 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 page 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** command 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 is allowed to issue this show command.

#### **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:

- 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.
- 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 certificate**

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 of 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 4-1 describes strings within the **show ca certificate** sample output.

**Table 4-1** *show ca certificate 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

#### **show ca crl**

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 of the **show ca crl** command. See Table 4-1 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

```

#### **show ca mypubkey rsa**

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 of 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

```

## 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. (Configuration Mode.)

**ca generate rsa key** *modulus*

### Syntax Description

<b>ca generate rsa key</b>	Generates an RSA key for the PIX Firewall.
<i>modulus</i>	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. Default domain is *ciscopix.com*.

### 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 communications protocol SSL to communicate with the PIX Firewall.

SSL uses the private key generated with the **ca generate rsa** command. For a certificate, SSL uses the one 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**

This 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
```

## clear Commands

Remove commands from the configuration or reset command values. (All modes.)

Table 4-2, Table 4-3, and Table 4-4 list each mode in which the **clear** commands first appear. Each **clear** command listed in one mode can be also accessed in each subsequent more secure mode going from unprivileged to configuration mode, but not from less secure modes.

**Table 4-2 Unprivileged Mode Clear Commands**

Clear Command	Description	Described on Command Page
<b>clear pager</b>	Resets the number of displayed lines to 24.	<a href="#">pager</a>

**Table 4-3 Privileged Mode Clear Commands**

Clear Command	Description	Described on Command Page
<b>clear arp</b>	Clears the ARP table.	<a href="#">arp</a>
<b>clear auth-prompt</b>	Removes an <b>auth-prompt</b> command statement from the configuration.	<a href="#">auth-prompt</a>
<b>clear blocks</b>	Resets the <b>show blocks</b> command statement counters.	<a href="#">show blocks/clear blocks</a>
<b>clear configure</b>	Resets command parameters in the configuration to their default values.	<a href="#">configure</a>
<b>clear flashfs</b>	Clears Flash memory prior to downgrading the PIX Firewall software version.	<a href="#">fragment</a>
<b>clear floodguard</b>	Removes Flood Defender which protects against flood attacks from configuration.	<a href="#">floodguard</a>
<b>clear local-host</b>	Resets the information displayed for the <b>show local-host</b> command.	<a href="#">local-host (clear and show)</a>
<b>clear passwd</b>	Resets the Telnet password back to "cisco."	<a href="#">passwd</a>
<b>clear traffic</b>	Resets the counters for the <b>show traffic</b> command.	<a href="#">show traffic/clear traffic</a>

**Table 4-3 Privileged Mode Clear Commands (continued)**

Clear Command	Description	Described on Command Page
<b>clear uauth</b>	Deletes one user's or all users' AAA authorization caches, which forces the users to reauthenticate the next time they create a connection.	<a href="#">uauth (clear and show)</a>
<b>clear xlate</b>	Clears the contents of the translation slots.	<a href="#">xlate (clear and show)</a>

**Table 4-4 Configuration Mode Clear Commands**

Clear Command	Description	Described on Command Page
<b>clear aaa</b>	Removes <b>aaa</b> command statements from the configuration.	<a href="#">aaa</a>
<b>clear aaa-server</b>	Removes <b>aaa-server</b> command statements from the configuration.	<a href="#">aaa-server</a>
<b>clear access-list</b>	Removes <b>access-list</b> command statements from the configuration. This command also stops all traffic through the PIX Firewall on the affected <b>access-list</b> command statements.	<a href="#">access-list</a>
<b>clear access-group</b>	Removes <b>access-group</b> command statements from the configuration.	<a href="#">access-group</a>
<b>clear alias</b>	Removes <b>alias</b> command statements from the configuration.	<a href="#">alias</a>
<b>clear apply</b>	Removes <b>apply</b> command statements from the configuration.	<a href="#">outbound/apply</a>
<b>clear conduit</b>	Removes <b>conduit</b> command statements from the configuration.	<a href="#">conduit</a>
<b>clear dhcpd</b>	Removes <b>dhcpd</b> command statements from the configuration.	<a href="#">dhcpd</a>
<b>clear established</b>	Removes <b>established</b> command statements from the configuration.	<a href="#">established</a>
<b>clear filter</b>	Removes <b>filter</b> command statements from the configuration.	<a href="#">filter</a>
<b>clear fixup</b>	Resets <b>fixup protocol</b> command statements to their default values.	<a href="#">fixup protocol</a>
<b>clear flashfs</b>	Clears Flash memory before downgrading to a previous PIX Firewall version.	<a href="#">fragment</a>
<b>clear global</b>	Removes <b>global</b> command statements from the configuration.	<a href="#">global</a>
<b>clear http</b>	Removes all HTTP hosts and disables the server.	<a href="#">http</a>
<b>clear icmp</b>	Removes <b>icmp</b> command statements from the configuration.	<a href="#">icmp</a>

Table 4-4 Configuration Mode Clear Commands (continued)

Clear Command	Description	Described on Command Page
<b>clear ip</b>	Sets all PIX Firewall interface IP addresses to 127.0.0.1 and stops all traffic.	<a href="#">ip address</a>
<b>clear ip address</b>	Clears all PIX Firewall interface IP addresses (configuration mode).	<a href="#">ip address</a>
<b>clear ip audit</b>	Clears IDS signature of interface (configuration mode).	<a href="#">ip audit</a>
<b>clear ip local pool</b>	Clears pool of local IP addresses for dynamic assignment to a VPN.	<a href="#">ip local pool</a>
<b>clear ip verify reverse-path</b>	Clears RPF IP spoofing protection (configuration mode).	<a href="#">ip verify reverse-path</a>
<b>clear [crypto] dynamic-map</b>	Remove <b>crypto dynamic-map</b> command statements from the configuration. The keyword <b>crypto</b> is optional.	<a href="#">crypto dynamic-map</a> and <a href="#">dynamic-map</a>
<b>clear [crypto] ipsec sa</b>	Delete the active IPSec security associations. The keyword <b>crypto</b> is optional.	<a href="#">crypto ipsec</a>
<b>clear [crypto] ipsec sa counters</b>	Clear the traffic counters maintained for each security association. The keyword <b>crypto</b> is optional.	<a href="#">crypto ipsec</a>
<b>clear [crypto] ipsec sa entry</b> <i>destination-address protocol spi</i>	Delete the active IPSec security association with the specified address, protocol, and SPI. The keyword <b>crypto</b> is optional.	<a href="#">crypto ipsec</a>
<b>clear [crypto] ipsec sa map</b> <i>map-name</i>	Delete the active IPSec security associations for the named crypto map set. The keyword <b>crypto</b> is optional.	<a href="#">crypto ipsec</a>
<b>clear [crypto] ipsec sa peer</b>	Delete the active IPSec security associations for the specified peer. The keyword <b>crypto</b> is optional.	<a href="#">crypto ipsec</a>
<b>clear [crypto] isakmp sa</b>	Delete the active IKE security associations. The keyword <b>crypto</b> is optional.	<a href="#">isakmp</a>
<b>clear [crypto] map</b>	Delete all parameters entered through the <b>crypto map</b> command belonging to the specified map. Does not delete dynamic maps.	<a href="#">crypto map</a>
<b>clear ipsec</b>	Remove <b>ipsec</b> command statements from the configuration.	<a href="#">ipsec</a>
<b>clear isakmp</b>	Remove <b>isakmp</b> command statements from the configuration.	<a href="#">isakmp</a>
<b>clear interface</b>	Clear counters for the <b>show interface</b> command.	<a href="#">interface</a>
<b>clear logging</b>	Clear syslog message queue accumulated by the <b>logging buffered</b> command.	<a href="#">logging</a>
<b>clear names</b>	Removes <b>name</b> command statements from the configuration.	<a href="#">name/names</a>
<b>clear nameif</b>	Reverts <b>nameif</b> command statements to default interface names and security levels.	<a href="#">nameif</a>

Table 4-4 Configuration Mode Clear Commands (continued)

Clear Command	Description	Described on Command Page
<b>clear nat</b>	Removes <b>nat</b> command statements from the configuration.	<a href="#">nat</a>
<b>clear outbound</b>	Removes <b>outbound</b> command statements from the configuration.	<a href="#">outbound/apply</a>
<b>clear pdm</b>	Removes all locations, disables logging and clears the PDM buffer. Internal PDM command.	<a href="#">pdm</a>
<b>clear rip</b>	Removes <b>rip</b> command statements from the configuration.	<a href="#">rip</a>
<b>clear route</b>	Removes <b>route</b> command statements from the configuration that do not contain the CONNECT keyword.	<a href="#">route</a>
<b>clear service</b>	Removes <b>service</b> command statements from the configuration.	<a href="#">service</a>
<b>clear snmp-server</b>	Removes <b>snmp-server</b> command statements from the configuration.	<a href="#">snmp-server</a>
<b>clear ssh</b>	Removes <b>ssh</b> command statement from the configuration.	<a href="#">ssh</a>
<b>clear static</b>	Removes <b>static</b> command statements from the configuration.	<a href="#">static</a>
<b>clear sysopt</b>	Removes <b>sysopt</b> command statements from the configuration.	<a href="#">sysopt</a>
<b>clear telnet</b>	Removes <b>telnet</b> command statements from the configuration.	<a href="#">telnet</a>
<b>clear tftp-server</b>	Removes <b>tftp-server</b> command statements from the configuration.	<a href="#">tftp-server</a>
<b>clear timeout</b>	Resets <b>timeout</b> command durations to their default values.	<a href="#">timeout</a>
<b>clear url-cache</b>	Removes <b>url-cache</b> command statements from the configuration.	<a href="#">url-cache</a>
<b>clear url-server</b>	Removes <b>url-server</b> command statements from the configuration.	<a href="#">url-server</a>
<b>clear virtual</b>	Removes <b>virtual</b> command statements from the configuration.	<a href="#">virtual</a>
<b>clear vpdn</b>	Removes <b>vpdn</b> command statements from the configuration.	<a href="#">vpdn</a>

# clock

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

**clock**

**clock set** *hh:mm:ss month day year*

**clock set** *hh:mm:ss day month year*

**show clock**

## Syntax Description

<i>hh:mm:ss</i>	The current hour:minutes:seconds expressed in 24-hour time; for example, <b>20:54:00</b> for 8:54 pm. Zeros can be entered as a single digit; for example, <b>21:0:0</b> .
<i>month</i>	The current month expressed as the first three characters of the month; for example, <b>apr</b> for April.
<i>day</i>	The current day of the month; for example, <b>1</b> .
<i>year</i>	The current year expressed as four digits; for example, <b>2000</b> .

## Usage Guidelines

The **clock** command lets you specify the current 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 current time with the **clock** or the **show clock** command.



### Note

The lifetime of a certificate and the certificate revocation list (CRL) is checked in GMT. If you are using IPSec with certificates, set the PIX Firewall clock to GMT timezone 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**.

A time prior to January 1, 1998 or after December 31, 2097 will not be accepted (the maximum date that the **clock** command can work to).

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 pm 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.

# conduit

Add, delete, or show conduits through the PIX Firewall for incoming connections. (Configuration mode.)

```
conduit permit | deny protocol global_ip global_mask [operator port [port]] foreign_ip
foreign_mask [operator port [port]]
```

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

```
conduit permit | deny icmp global_ip global_mask foreign_ip foreign_mask [icmp_type]
```

```
clear conduit
```

```
show conduit
```

## Syntax Description

<b>permit</b>	Permit access if the conditions are matched.
<b>deny</b>	Deny access if the conditions are matched.
<i>protocol</i>	Specify the transport protocol for the connection. Possible literal values are <b>icmp</b> , <b>tcp</b> , <b>udp</b> , or an integer in the range 0 through 255 representing an IP protocol number. Use <b>ip</b> to specify all transport protocols. You can view valid protocol numbers online at the following website:  <a href="http://www.isi.edu/in-notes/iana/assignments/protocol-numbers">http://www.isi.edu/in-notes/iana/assignments/protocol-numbers</a>  If you specify the <b>icmp</b> protocol, you can permit or deny ICMP access to one or more global IP addresses. Specify the ICMP type in the <i>icmp_type</i> variable, or omit to specify all ICMP types. See "Usage Guidelines" for a complete list of the ICMP types.
<i>global_ip</i>	A global IP address previously defined by a <b>global</b> or <b>static</b> command. You can use <b>any</b> if the <i>global_ip</i> and <i>global_mask</i> are 0.0.0.0 0.0.0.0. The <b>any</b> option applies the <b>permit</b> or <b>deny</b> parameters to the global addresses.  If <i>global_ip</i> is a host, you can omit <i>global_mask</i> by specifying the <b>host</b> command before <i>global_ip</i> .  For example:  <pre><b>conduit permit tcp host 209.165.201.1 eq ftp any</b></pre> This example lets any foreign host access global address 209.165.201.1 for FTP.
<i>global_mask</i>	Network mask of <i>global_ip</i> . The <i>global_mask</i> 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 <b>0</b> for <i>global_ip</i> , use <b>0</b> for the <i>global_mask</i> ; otherwise, enter the <i>global_mask</i> appropriate to <i>global_ip</i> .

---

<i>foreign_ip</i>	<p>An external IP address (host or network) that can access the <i>global_ip</i>. You can specify <b>0.0.0.0</b> or <b>0</b> for any host. If both the <i>foreign_ip</i> and <i>foreign_mask</i> are 0.0.0.0 0.0.0.0, you can use the shorthand <b>any</b> option.</p> <p>If <i>foreign_ip</i> is a host, you can omit <i>foreign_mask</i> by specifying the <b>host</b> command before <i>foreign_ip</i>.</p> <p>For example:</p> <pre>conduit permit tcp any eq ftp host 209.165.201.2</pre> <p>This example lets foreign host 209.165.201.2 access any global address for FTP.</p>
<i>foreign_mask</i>	<p>Network mask of <i>foreign_ip</i>. The <i>foreign_mask</i> 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 <b>0</b> for <i>foreign_ip</i>, use <b>0</b> for the <i>foreign_mask</i>; otherwise, enter the <i>foreign_mask</i> appropriate to <i>foreign_ip</i>. You can also specify a mask for subnetting.</p> <p>For example: 255.255.255.192.</p>
<i>operator</i>	<p>A comparison operand that allows you to specify a port or a port range.</p> <p>Use without an operator and port to indicate all ports; for example:</p> <pre>conduit permit tcp any any</pre> <p>Use <b>eq</b> and a port to permit or deny access to just that port. For example use <b>eq ftp</b> to permit or deny access only to FTP:</p> <pre>conduit deny tcp host 192.168.1.1 eq ftp 209.165.201.1</pre> <p>Use <b>lt</b> and a port to permit or deny access to all ports less than the port you specify. For example, use <b>lt 2025</b> to permit or deny access to the well known ports (1 to 1024).</p> <pre>conduit permit tcp host 192.168.1.1 lt 1025 any</pre> <p>Use <b>gt</b> and a port to permit or deny access to all ports greater than the port you specify. For example, use <b>gt 42</b> to permit or deny ports 43 to 65535.</p> <pre>conduit deny udp host 192.168.1.1 gt 42 host 209.165.201.2</pre> <p>Use <b>neq</b> and a port to permit or deny access to every port except the ports that you specify. For example, use <b>neq 10</b> to permit or deny ports 1-9 and 11 to 65535:</p> <pre>conduit deny tcp host 192.168.1.1 neq 10 host 209.165.201.2 neq 42</pre> <p>Use <b>range</b> and a port range to permit or deny access to only those ports named in the range. For example, use <b>range 10 1024</b> to permit or deny access only to ports 10 through 1024. All other ports are unaffected.</p> <pre>conduit deny tcp any range ftp telnet any</pre> <p>By default, all ports are denied until explicitly permitted.</p>

---

---

<i>port</i>	<p>Service(s) you permit to be used while accessing <i>global_ip</i> or <i>foreign_ip</i>. Specify services by the port that handles it, such as <b>smtp for port 25</b>, <b>www</b> 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.</p> <p>For example:</p> <pre><b>conduit deny tcp any any</b></pre> <p>This command is the default condition for the <b>conduit</b> command in that all ports are denied until explicitly permitted.</p> <p>You can view valid port numbers online at the following website:</p> <p><a href="http://www.isi.edu/in-notes/iana/assignments/port-numbers">http://www.isi.edu/in-notes/iana/assignments/port-numbers</a></p> <p>See “Ports” in Chapter 1, “Using PIX Firewall Commands” for a list of valid port literal names in port ranges; for example, <b>ftp h323</b>. You can also specify numbers.</p>
<i>icmp_type</i>	<p>The type of ICMP message. Table 4-5 lists the ICMP type literals that you can use in this command. Omit this option to mean all ICMP types. An example of this command that permits all ICMP types is <b>conduit permit icmp any any</b>. This command lets ICMP pass inbound and outbound.</p>

---

### Usage Guidelines

A **conduit** command statement creates an exception to the PIX Firewall Adaptive Security Algorithm by permitting connections from one PIX Firewall network interface to access hosts on another.

The **clear conduit** command removes all **conduit** command statements from your configuration.

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.



#### Note

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.

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.

### 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_port [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_port [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 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. Use the **show conduit** command to view the **conduit** command statements in the configuration and the number of times (hit count) an element has been matched during a **conduit** command search.

If you prefer more selective ICMP access, you can specify a single ICMP message type as the last option in this command. [Table 4-5](#) lists possible ICMP types values.

**Table 4-5 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-reply
14	timestamp-request
15	information-request
16	information-reply
17	mask-request
18	mask-reply

**Table 4-5 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 InternetPhone 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 InternetPhone 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

Clear or merge the current configuration with that on floppy or Flash memory, start configuration mode, or view current configuration.


**Note**

The PIX 506, PIX 515, and PIX 525 do not support use of the **configure floppy** command.

**clear configure primary | secondary | all**

**configure net** [[*server\_ip*]:*filename*]

**configure floppy**

**configure memory**

**configure terminal**

**show configure**

**Syntax Description**

<b>clear</b>	Clears aspects of the current configuration in RAM. Use the <b>write erase</b> command to clear the complete configuration.
<b>primary</b>	Sets the <b>interface</b> , <b>ip</b> , <b>mtu</b> , <b>nameif</b> , and <b>route</b> commands to their default values. In addition, interface names are removed from all commands in the configuration.
<b>secondary</b>	Removes the <b>aaa-server</b> , <b>alias</b> , <b>access-list</b> , <b>apply</b> , <b>conduit</b> , <b>global</b> , <b>outbound</b> , <b>static</b> , <b>telnet</b> , and <b>url-server</b> command statements from your configuration.
<b>net</b>	Loads the configuration from a TFTP server and the path you specify. Privileged mode.
<b>all</b>	Combines the <b>primary</b> and <b>secondary</b> options.
<b>floppy</b>	Merges the current configuration with that on diskette.
<b>memory</b>	Merges the current configuration with that in Flash memory.
<b>terminal</b>	Starts configuration mode to enter configuration commands from a terminal. Exit configuration mode by entering the <b>quit</b> command. Configuration mode.
<i>server_ip</i>	Merges the current configuration with that available across the network at another location, which is defined with the <b>tftp-server</b> 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 <b>tftp-server</b> command, do not specify it in the <b>configure</b> command; instead just use a colon (:) without a filename.

**Usage Guidelines**

The **clear configure** 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 the **clear configure** command. The **clear configure secondary** command does not prompt you before deleting lines from your configuration.

---

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 *: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. This does not occur if you are using Cisco TFTP Server version 1.1 for Windows NT.

**Note**

---

Many TFTP servers require the configuration file to be world-readable to be accessible.

---

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.

The **configure memory** command merges the configuration in Flash memory into the current configuration in RAM.

The **configure terminal** command starts configuration mode. 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. Use the **write terminal** command to display the current configuration.

The **show configure** command lists the contents of the configuration in Flash memory.

Each command statement from diskette (with **configure floppy**), Flash memory (with **configure memory**), or TFTP transfer (with **configure net**) 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 on diskette or Flash memory is identical to an existing command in the current configuration, it is ignored.
- If the command on diskette or Flash memory 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 host name when that command is read from diskette.

---

**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
```

# copy tftp flash

Change software images without requiring access to the TFTP monitor mode. (Configuration mode.)

```
copy tftp:[[//location] [/pathname]] flash:[image | pdm]
```

Syntax Description	copy tftp flash	Download Flash memory software images via TFTP without using monitor mode.
	<i>location</i>	Either an IP address or a name that resolves to an IP address via the PIX Firewall naming resolution mechanism.
	<i>pathname</i>	PIX Firewall must know how to reach this location via its routing table information. This information is determined by the <b>ip address</b> command, the <b>route</b> 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.
	<b>image</b>	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).
	<b>pdm</b>	Download the selected PDM image files to Flash memory. These files are available to the PIX Firewall immediately, without a reboot.

## Usage Guidelines

The **copy tftp flash** command allows you to download a software image via TFTP. You can use the **copy tftp flash** command with any PIX Firewall model running version 5.1 or later.

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
```

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.

For example, if you want to download the `pix512.bin` file from the D: partition on a Windows system (IP address 10.1.1.5), you would access the Cisco TFTP Server **View>Options** menu and enter the filename path in the **TFTP server root directory** edit box; for example, `D:\pix_images`. To copy the file to the PIX Firewall, use the following **copy tftp** command.

```
copy tftp://10.1.1.5/pix512.bin flash
```

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.

**Note**


---

Images prior to version 5.1 cannot be retrieved using this mechanism.

---

**Examples**

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.

```
tftp-server outside 10.1.1.5 pix512.bin
Warning: 'outside' interface has a low security level (0).
write memory
Building configuration...
Cryptochecksum: 017c452b d54be501 8620ba48 490f7e99
[OK]
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
```

# crypto dynamic-map

Create, view, or delete a dynamic crypto map entry. (Configuration mode.)

**crypto dynamic-map** *dynamic-map-name* *dynamic-seq-num*

**no crypto dynamic-map** *dynamic-map-name* [*dynamic-seq-num*]

**crypto dynamic-map** *dynamic-map-name* *dynamic-seq-num* **match address** *acl\_name*

**no crypto dynamic-map** *dynamic-map-name* *dynamic-seq-num* **match address** *acl\_name*

**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 peer** *hostname* | *ip-address*

**crypto dynamic-map** *dynamic-map-name* *dynamic-seq-num* **set pfs** [**group1** | **group2**]

**no crypto dynamic-map** *dynamic-map-name* *dynamic-seq-num* **set pfs**

**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 security-association lifetime**  
**seconds** | **kilobytes**

**crypto dynamic-map** *dynamic-map-name* *dynamic-seq-num* **set transform-set**  
*transform-set-name1* [... *transform-set-name9*]

**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

<i>dynamic-map-name</i>	Specify the name of the dynamic crypto map set.
<i>dynamic-seq-num</i>	Specify the sequence number that corresponds to the dynamic crypto map entry.
<i>subcommand</i>	Various subcommands ( <b>match address</b> , <b>set transform-set</b> , and so on).
<b>tag</b> <i>map-name</i>	(Optional) Show the crypto dynamic map set with the specified <i>map-name</i> .

**Note**

The **crypto dynamic-map** subcommands, such as **match address**, **set peer**, and **set pfs** are described in the **crypto map** command page. 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. command page. See this command page for the descriptions of these commands, including syntax descriptions.

**Usage Guidelines**

The sections that follow describe each **crypto dynamic-map** command.

**crypto dynamic-map**

The **crypto dynamic-map** command allows you to 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** removes 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 allows you to 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 allows you to 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 allows you to 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 **dynamic crypto map** command statements are used for determining whether or not traffic should be protected.

**Note**

The only parameter required in a **dynamic crypto 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.

```
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 for the **show 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 above **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
crypto dynamic-map dyn1 set transform-set tauth t1
crypto dynamic-map dyn1 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
```

**crypto dynamic-map match address**

See the [crypto map match address](#) command within the [crypto map](#) command page for information about this command.

**crypto dynamic-map set peer**

See the [crypto map set peer](#) command within the [crypto map](#) command page for information about this command.

**crypto dynamic-map set pfs**

See the [crypto map set pfs](#) command within the [crypto map](#) command page 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 page for information about this command.

**crypto dynamic-map set transform-set**

See the [crypto map set transform-set](#) command within the [crypto map](#) command page 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. (Configuration mode.)

**crypto ipsec security-association lifetime seconds** *seconds* | **kilobytes** *kilobytes*

**no crypto ipsec security-association lifetime seconds** | **kilobytes**

**show crypto ipsec security-association lifetime**

**crypto ipsec transform-set** *transform-set-name transform1* [*transform2* [*transform3*]]

**no crypto ipsec transform-set** *transform-set-name*

**show crypto ipsec transform-set** [**tag** *transform-set-name*]

**crypto ipsec transform-set** *transform-set-name* **mode transport**

**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 sa** [**map** *map-name* | **address** | **identity**] [**detail**]

Syntax Description		
<b>address</b>	(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).	
<i>destination-address</i>	Specify the IP address of your peer or the remote peer.	
<b>detail</b>	(Optional) Show detailed error counters. (The default is the high level send/receive error counters.)	
<b>identity</b>	(Optional) Show only the flow information. It does not show the security association information.	
<i>interface-name</i>	Specify the identifying interface (outside or external) to be used by the PIX Firewall to identify itself to remote 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.	
<i>ip-address</i>	Specify a remote peer's IP address.	
<b>ipsec-isakmp</b>	Indicate that IKE will be used to establish the IPSec security associations for protecting the traffic specified by this crypto map entry.	

<b>ipsec-manual</b>	Indicate that IKE will not be used to establish the IPsec security associations for protecting the traffic specified by this crypto map entry.
<b>kilobytes</b> <i>kilobytes</i>	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).
<b>map</b> <i>map-name</i>	The name of the crypto map set.
<i>peer-name</i>	Specify a remote peer's name as the fully qualified domain name. For example, remoteppeer.example.com.
<i>protocol</i>	Specify either the AH or ESP protocol.
<b>mode</b> <i>transport</i>	Specifies the transform set to accept transport mode requests in addition to the tunnel mode request.
<b>seconds</b> <i>seconds</i>	Specify the number of seconds a security association will live before it expires. The default is 28,800 seconds (eight hours).
<i>seq-num</i>	The number you assign to the crypto map entry.
<i>spi</i>	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 hexadecimal value of FFFF FFFF).
<b>tag</b> <i>transform-set-name</i>	(Optional) Show only the transform sets with the specified transform-set-name.
<i>transform1</i> <i>transform2</i> <i>transform3</i>	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.
<i>transform-set-name</i>	Specify the name of the transform set to create or modify.

## Usage Guidelines

The sections that follow describe each **crypto ipsec** 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 allows you to 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.

---

## Examples

This example shortens both lifetimes, because the administrator feels there is a higher risk that the keys could be compromised. The timed 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 for 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
```

### **crypto ipsec transform-set**

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.

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 could specify the AH protocol, the ESP protocol, or both. 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, see Chapter 4, “Basic VPN Configuration” in the *Cisco PIX Firewall and VPN Configuration Guide*.

This example defines one transform set (named “standard”), which will be 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
```

#### **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.

#### **clear [crypto] ipsec sa**

The **clear [crypto] ipsec sa** command allows you 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**, or **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 access-list or peers, the **clear [crypto] ipsec sa** command will not be enough to activate 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 re initializes 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 allows you to 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 of 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.

The following is sample output for the **show crypto ipsec sa** command:

```

show crypto ipsec sa

interface: outside
  Crypto map tag: firewall-alice, 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-alice
      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-alice
      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-alice, 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-alice
    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-alice
    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. (Configuration mode.)



## Note

---

PIX 506 supports 25 VPN peers/ISAKMP peers.

---

**crypto map** *map-name* **client** [**token**] **authentication** *aaa-server-name*

**no crypto map** *map-name* **client** [**token**] **authentication** *aaa-server-name*

**crypto map** *map-name* **client configuration** **address** **initiate** | **respond**

**no crypto map** *map-name* **client configuration** **address** **initiate** | **respond**

**crypto map** *map-name* **interface** *interface-name*

**no crypto map** *map-name* **interface** *interface-name*

**show crypto map** [**interface** *interface-name* | **tag** *map-name*]

**clear crypto map** *map-name*

**crypto map** *map-name* *seq-num* **ipsec-isakmp** | **ipsec-manual** [**dynamic** *dynamic-map-name*]

**no crypto map** *map-name* *seq-num*

**crypto map** *map-name* *seq-num* **match** **address** *acl\_name*

**no crypto map** *map-name* *seq-num* **match** **address** *acl\_name*

**crypto map** *map-name* *seq-num* **set peer** *hostname* | *ip-address*

**no crypto map** *map-name* *seq-num* **set peer** *hostname* | *ip-address*

**crypto map** *map-name* *seq-num* **set pfs** [**group1** | **group2**]

**no crypto map** *map-name* *seq-num* **set pfs**

**crypto map** *map-name* *seq-num* **set security-association** **lifetime** **seconds** *seconds* |  
**kilobytes** *kilobytes*

**no crypto map** *map-name* *seq-num* **set security-association** **lifetime** **seconds** *seconds* |  
**kilobytes** *kilobytes*

**crypto map** *map-name* **set session-key inbound | outbound ah** *spi hex-key-string*

**no crypto map** *map-name seq-num set session-key inbound | outbound ah*

**crypto map** *map-name set session-key inbound | outbound esp spi cipher hex-key-string*  
**[authenticator hex-key-string]**

**no crypto map** *map-name seq-num set session-key inbound | outbound esp*

**crypto map** *map-name seq-num set transform-set transform-set-name1*  
**[... transform-set-name6]**

**no crypto map** *map-name seq-num set transform-set transform-set-name1*  
**[... transform-set-name6]**

#### Syntax Description

<b>map</b> <i>map-name</i>	The name of the crypto map set.
<i>aaa-server-name</i>	The name of the AAA server that will authenticate the user during IKE authentication. The two AAA server options available are TACACS+ and RADIUS.
<b>token</b>	Indicate a token-based server for user authentication is used.
<b>initiate</b>	Indicate that the PIX Firewall will attempt to set IP addresses for each peer.
<b>respond</b>	Indicate that the PIX Firewall will accept requests for IP addresses from any requesting peer.
<b>interface</b> <i>interface-name</i>	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.
<b>tag</b> <i>map-name</i>	(Optional) Show the crypto map set with the specified map name.
<i>seq-num</i>	The number you assign to the crypto map entry.
<b>ipsec-isakmp</b>	Indicate that IKE will be used to establish the IPsec security associations for protecting the traffic specified by this crypto map entry.
<b>ipsec-manual</b>	Indicate that IKE will not be used to establish the IPsec security associations for protecting the traffic specified by this crypto map entry.
<b>dynamic</b>	(Optional) Specify that this crypto map entry is to reference a pre-existing dynamic crypto map.
<i>dynamic-map-name</i>	(Optional) Specify the name of the dynamic crypto map set to be used as the policy template.
<i>acl_name</i>	Identify the named encryption access list. This name should match the name argument of the named encryption access list being matched.
<b>match address</b>	Specify an access list for a crypto map entry.
<b>set peer</b>	Specify an IPsec peer in a crypto map entry.

<i>hostname</i>	Specify a peer by its host name. This is the peer's host name concatenated with its domain name. For example, myhost.example.com.
<i>ip-address</i>	Specify a peer by its IP address.
<b>set pfs</b>	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.)
<b>group1</b>	Specify that IPSec should use the 768-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
<b>group2</b>	Specify that IPSec should use the 1024-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
<b>set security-association lifetime</b>	Set the lifetime a security association will last in either seconds or kilobytes. For use with either <b>seconds</b> or <b>kilobyte</b> keywords.
<b>seconds</b> <i>seconds</i>	Specify the number of seconds a security association will live before it expires. The default is 28,800 seconds (eight hours).
<b>kilobytes</b> <i>kilobytes</i>	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.
<b>set session-key</b>	Manually specify the IPSec session keys within a crypto map entry.
<b>inbound</b>	Set the inbound IPSec session key. (You must set both inbound and outbound keys.)
<b>outbound</b>	Set the outbound IPSec session key. (You must set both inbound and outbound keys.)
<b>ah</b>	Set the IPSec session key for the AH protocol. Specify <b>ah</b> when the crypto map entry's transform set includes an AH transform. AH protocol provides authentication via MD5-HMAC and SHA-HMAC.
<i>spi</i>	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 hexadecimal 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.
<i>hex-key-string</i>	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: <ul style="list-style-type: none"> <li>• DES algorithm, specify at least 16 hexadecimal digits per key.</li> <li>• MD5 algorithm, specify at least 32 hexadecimal digits per key.</li> <li>• SHA algorithm, specify 40 hexadecimal digits per key.</li> </ul> Longer key sizes are simply hashed to the appropriate length.

<b>esp</b>	Set the IPsec session key for the ESP protocol. Specify <b>esp</b> 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.
<b>cipher</b>	Indicate that the key string to use with the ESP encryption transform.
<b>authenticator</b>	(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.
<b>set transform-set</b>	Specify which transform sets can be used with the crypto map entry.
<i>transform-set-name</i>	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.
<i>transform1</i> <i>transform2</i> <i>transform3</i>	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.

### Usage Guidelines

The sections that follow describe each **crypto map** command.

#### crypto map client authentication

The **crypto map client authentication** command enables the Extended Authentication (Xauth) feature, which lets you prompt for a TACACS+/RADIUS username and password during IKE authentication. You must first have your basic AAA server set up to use this feature.

This command tells the PIX Firewall during Phase 1 of IKE to use the Xauth (RADIUS/TACACS+) 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

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.

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 3.0  
Cisco VPN 3000 Client, version 2.5 or later  
Cisco Secure VPN Client, version 1.1 or later

## 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
```

### crypto map client configuration address

Use the **crypto map client configuration address** command to configure IKE Mode Configuration on your PIX Firewall. The IKE Mode Configuration allows the PIX Firewall to download an IP address to the remote peer (client) as part of an IKE negotiation. With **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. The 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 later supports the IKE Mode Configuration.

See Chapter 4, "Basic VPN Configuration" of 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**

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 for the **show crypto map** command:

```
show crypto map
```

```
Crypto Map: "firewall-alice" pif: outside local address: 172.21.114.123
```

```
Crypto Map "firewall-alice" 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-alice 10 ipsec-isakmp
crypto map firewall-alice 10 set peer 172.21.114.67
crypto map firewall-alice 10 set transform-set t1
crypto map firewall-alice 10 match address 141
```

The following is sample output for 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: 010203040506070809010203040506070809010203040506070809,
  Outbound esp spi: 0
    cipher key: ,
    auth key: ,
  Outbound ah spi: 256,
    key: 010203040506070809010203040506070809010203040506070809,
```

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
010203040506070809010203040506070809010203040506070809
crypto map multi-peer 20 set session-key outbound ah 256
010203040506070809010203040506070809010203040506070809
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.

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 98765432109876549876543210987654
crypto map mymap 10 set session-key outbound ah 256 fedcbafedcbafedcbafedcbafedcbafedc
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.

### Examples

```
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), while 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.

This 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.



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 0000111122223333444455556666777788889999
crypto map mymap 10 set session-key outbound esp 300 cipher abcdefabcdefabcd
authenticator 9999888877776666555544443333222211110000
```

#### 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.

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.



