



## Configuring IPsec and Certification Authorities

This chapter provides information about using IP Security Protocol (IPsec), Internet Key Exchange (IKE), and certification authority (CA) technology with the PIX Firewall.

This chapter includes the following sections:

- [How IPsec Works, page 6-1](#)
- [Internet Key Exchange \(IKE\), page 6-2](#)
- [Using Certification Authorities, page 6-7](#)
- [Configuring IPsec, page 6-13](#)
- [Using Dynamic Crypto Maps, page 6-22](#)
- [Manual Configuration of SAs, page 6-25](#)
- [Viewing IPsec Configuration, page 6-28](#)
- [Clearing SAs, page 6-28](#)

### How IPsec Works

IPsec provides authentication and encryption services to protect unauthorized viewing or modification of data within your network or as it is transferred over an unprotected network, such as the public Internet. IPsec is generally implemented in two types of configurations:

- **Site-to-site**—This configuration is used between two IPsec security gateways, such as PIX Firewall units. A site-to-site VPN interconnects networks in different geographic locations. For information that is specific for configuring IPsec in this configuration, refer to [Chapter 7, “Site-to-Site VPN Configuration Examples.”](#)
- **Remote access**—This configuration is used to allow secure remote access for VPN clients, such as mobile users. A remote access VPN allows remote users to securely access centralized network resources. For information that is specific for configuring IPsec in this configuration, refer to [Chapter 8, “Managing VPN Remote Access.”](#)

Two different security protocols are included within the IPsec standard:

- **Encapsulating Security Payload (ESP)**—Provides authentication, encryption, and anti-replay services.
- **Authentication Header (AH)**—Provides authentication and anti-replay services.

IPSec can be configured to work in two different modes:

- **Tunnel Mode**—This is the normal way in which IPSec is implemented between two PIX Firewall units (or other security gateways) that are connected over an untrusted network, such as the public Internet.
- **Transport Mode**—This method of implementing IPSec is typically done with L2TP to allow authentication of native Windows 2000 VPN clients. For information about configuring L2TP, refer to “[Using PPTP for Remote Access](#),” in [Chapter 8](#), “[Managing VPN Remote Access](#).”

The main task of IPSec is to allow the exchange of private information over an insecure connection. IPSec uses encryption to protect information from interception or eavesdropping. However, to use encryption efficiently, both parties should share a secret that is used for both encryption and decryption of the information.

IPSec operates in two phases to allow the confidential exchange of a shared secret:

- **Phase 1**, which handles the negotiation of security parameters required to establish a secure channel between two IPSec peers. Phase 1 is generally implemented through the Internet Key Exchange (IKE) protocol. If the remote IPSec peer cannot perform IKE, you can use manual configuration with pre-shared keys to complete Phase 1.
- **Phase 2**, which uses the secure tunnel established in Phase 1 to exchange the security parameters required to actually transmit user data.

The secure tunnels used in both phases of IPSec are based on security associations (SAs) used at each IPSec end point. SAs describe the security parameters, such as the type of authentication and encryption that both end points agree to use.

## Internet Key Exchange (IKE)

This section describes the Internet Key Exchange (IKE) protocol and how it works with IPSec to make VPNs more scalable. This section includes the following topics:

- [IKE Overview, page 6-2](#)
- [Configuring IKE, page 6-4](#)
- [Disabling IKE, page 6-6](#)
- [Using IKE with Pre-Shared Keys, page 6-6](#)

### IKE Overview

IKE is a protocol used by IPSec for completion of Phase 1. IKE negotiates and assigns SAs for each IPSec peer, which provide a secure channel for the negotiation of the IPSec SAs in Phase 2. IKE provides the following benefits:

- Eliminates the need to manually specify all the IPSec security parameters at both peers
- Lets you specify a lifetime for the IKE SAs
- Allows encryption keys to change during IPSec sessions
- Allows IPSec to provide anti-replay services
- Enables CA support for a manageable, scalable IPSec implementation
- Allows dynamic authentication of peers

IKE negotiations must be protected, so each IKE negotiation begins by each peer agreeing on a common (shared) IKE policy. This policy states the security parameters that will be used to protect subsequent IKE negotiations. After the two peers agree upon a policy, the security parameters of the policy are identified by an SA established at each peer, and these SAs apply to all subsequent IKE traffic during the negotiation.

There are five parameters to define in each IKE policy. These parameters apply to the IKE negotiations when the IKE SA is established. [Table 6-1](#) provides the five IKE policy keywords and their permitted values.

**Table 6-1 IKE Policy Keywords**

Keyword	Meaning	Description
des 3des	56-bit DES-CBC 168-bit Triple DES	Specifies the symmetric encryption algorithm used to protect user data transmitted between two IPSec peers. The default is 56-bit DES-CBC, which is less secure and faster than the alternatives.
aes aes-192 ase-256		The Advanced Encryption Standard is introduced with PIX Firewall version 6.3 and supports three different key lengths of 128, 192, 256 bytes.
sha md5	SHA-1 (HMAC variant) MD5 (HMAC variant)	Specifies the hash algorithm used to ensure data integrity. The default is SHA-1. MD5 has a smaller digest and is considered to be slightly faster than SHA-1. There has been a demonstrated successful (but extremely difficult) attack against MD5; however, the HMAC variant used by IKE prevents this attack.
rsa-sig pre-share	RSA signatures pre-shared keys	Specifies the method of authentication used to establish the identity of each IPSec peer. The default, RSA signatures, provide non-repudiation for the IKE negotiation (you can prove to a third party after the fact that you had an IKE negotiation with a specific peer). Pre-shared keys do not scale well with a growing network but are easier to set up in a small network.  For further information about the two authentication methods, refer to the following sections: <ul style="list-style-type: none"> <li>• <a href="#">“Using IKE with Pre-Shared Keys”</a></li> <li>• <a href="#">“Using Certification Authorities”</a></li> </ul>
1 2 5	Group 1 (768-bit Diffie-Hellman) Group 2 (1024-bit Diffie-Hellman) Group 5 (1536-bit Diffie-Hellman)	Specifies the Diffie-Hellman group identifier, which is used by the two IPSec peers to derive a shared secret without transmitting it to each other. The default, Group 1 (768-bit Diffie-Hellman) requires less CPU time to execute but is less secure than Group 2 (1024-bit Diffie-Hellman).  Support for Diffie-Hellman Group 5 is introduced with PIX Firewall Version 6.3.
integer value	120 to 86,400 seconds	Specifies the SA lifetime. The default is 86,400 seconds or 24 hours. As a general rule, a shorter lifetime (up to a point) provides more secure IKE negotiations. However, with longer lifetimes, future IPSec security associations can be set up more quickly.

There is an implicit trade-off between security and performance when you choose a specific value for each parameter. The level of security provided by the default values is adequate for the security requirements of most organizations. If you are interoperating with a peer that supports only one of the values for a parameter, your choice is limited to the other peer's supported value.

You can create multiple IKE policies, each with a different combination of parameter values. For each policy that you create, you assign a unique priority (1 through 65,534, with 1 being the highest priority). If you do not configure any policies, your PIX Firewall will use the default policy, which is always set to the lowest priority, and which contains each parameter's default value. If you do not specify a value for a specific parameter, the default value is assigned.

When the IKE negotiation begins, the peer that initiates the negotiation will send all its policies to the remote peer, and the remote peer will try to find a match. The remote peer checks each of its policies in order of its priority (highest priority first) until a match is found.

A match is made when both policies from the two peers contain the same encryption, hash, authentication, and Diffie-Hellman parameter values, and when the remote peer's policy specifies a lifetime less than or equal to the lifetime in the policy being compared. If the lifetimes are not identical, the shorter lifetime—from the remote peer's policy—will be used. If no acceptable match is found, IKE refuses negotiation and the IKE SA will not be established.

## Configuring IKE

To enable and configure IKE, perform the following steps:



### Note

If you do not specify a value for a given policy parameter, the default value is assigned.

**Step 1** Identify the policy to create. Each policy is uniquely identified by the priority number you assign.

```
isakmp policy priority
```

For example:

```
isakmp policy 20
```

**Step 2** Specify the encryption algorithm:

```
isakmp policy priority encryption aes | aes-192 | aes-256 | des | 3des
```

For example:

```
isakmp policy 20 encryption des
```

**Step 3** Specify the hash algorithm:

```
isakmp policy priority hash md5 | sha
```

For example:

```
isakmp policy 20 hash md5
```

**Step 4** Specify the authentication method:

```
isakmp policy priority authentication pre-share | rsa-sig
```

For example:

```
isakmp policy 20 authentication rsa-sig
```

For further information about the two authentication methods, refer to the following sections:

- [“Using IKE with Pre-Shared Keys”](#)
- [“Using Certification Authorities”](#)

**Step 5** Specify the Diffie-Hellman group identifier:

```
isakmp policy priority group 1 | 2 | 5
```




---

**Note** Support for Diffie-Hellman group 5 is introduced with PIX Firewall version 6.3

---

For example:

```
isakmp policy 20 group 2
```

**Step 6** Specify the security association’s lifetime:

```
isakmp policy priority lifetime seconds
```

For example:

```
isakmp policy 20 lifetime 5000
```

The following example shows two policies with policy 20 as the highest priority, policy 30 as the next priority, and the existing default policy as the lowest priority:

```
isakmp policy 20 encryption des
isakmp policy 20 hash md5
isakmp policy 20 authentication rsa-sig
isakmp policy 20 group 2
isakmp policy 20 lifetime 5000

isakmp policy 30 authentication pre-share
isakmp policy 30 lifetime 10000
```

In this example, the encryption des of policy 20 would not appear in the written configuration because this is the default for the encryption algorithm parameter.

**Step 7** (Optional) View all existing IKE policies:

```
show isakmp policy
```

The following is an example of the output after the policies 20 and 30 in the previous example were configured:

```
Protection suite priority 20
  encryption algorithm:  DES - Data Encryption Standard (56 bit keys)
  hash algorithm:        Message Digest 5
  authentication method: Rivest-Shamir-Adleman Signature
  Diffie-Hellman group:  #2 (1024 bit)
  lifetime:              5000 seconds, no volume limit
Protection suite priority 30
  encryption algorithm:  DES - Data Encryption Standard (56 bit keys)
  hash algorithm:        Secure Hash Standard
  authentication method: Pre-Shared Key
  Diffie-Hellman group:  #1 (768 bit)
  lifetime:              10000 seconds, no volume limit
```

```

Default protection suite
  encryption algorithm:  DES - Data Encryption Standard (56 bit keys)
  hash algorithm:       Secure Hash Standard
  authentication method: Rivest-Shamir-Adleman Signature
  Diffie-Hellman group:  #1 (768 bit)
  lifetime:             86400 seconds, no volume limit

```

**Note**


---

Although the output shows “no volume limit” for the lifetimes, you can currently only configure a time lifetime (such as 86,400 seconds) with IKE; volume limit lifetimes are not currently configurable.

---

## Disabling IKE

To disable IKE, you must make these concessions at the peers:

- All the IPSec security associations are manually specified in the crypto maps at all peers.
- IPSec security associations will never time out for a given IPSec session.
- The encryption keys never change during IPSec sessions between peers.
- Anti-replay services will not be available between the peers.
- CA support cannot be used.

To disable IKE, use the following command:

```
no crypto isakmp enable interface-name
```

For example:

```
no crypto isakmp enable outside
```

## Using IKE with Pre-Shared Keys

If you use the IKE authentication method of pre-shared keys, manually configure these keys on the PIX Firewall and its peer(s). You can specify the same key to share with multiple peers, but it is more secure to specify different keys to share between different pairs of peers. To configure a pre-shared key on the PIX Firewall, perform the following steps:

---

**Step 1** Configure the PIX Firewall host name:

```
hostname newname
```

For example:

```
hostname mypixfirewall
```

In this example, “mypixfirewall” is the name of a unique host in the domain.

When two peers use IKE to establish IPSec security associations, each peer sends its identity to its peer. Each peer’s identity is set either to its host name or its IP address. By default, the identity of the PIX Firewall is set to its IP address. If necessary, you can change the identity to be a host name instead. As a general rule, set all peers’ identities the same way—either all peers should use their IP addresses or all peers should use their host names. If some peers use their host names and some peers use their IP addresses to identify themselves to one another, IKE negotiations could fail if a peer’s identity is not recognized and a DNS lookup is unable to resolve the identity.

**Step 2** Configure the PIX Firewall domain name:

```
domain-name name
```

For example:

```
domain-name example.com
```

**Step 3** Specify the pre-shared key at the PIX Firewall:

```
isakmp key keystring address peer-address [netmask mask]
```

Replace *keystring* with the password string that the PIX Firewall and its peer will use for authentication. Replace *peer-address* with the remote peer's IP address.

For example:

```
isakmp key 1234567890 address 192.168.1.100
```

The pre-shared key is 1234567890, and the peer's address is 192.168.1.100.



---

**Note** Netmask lets you configure a single key to be shared among multiple peers. You would use the netmask of 0.0.0.0. However, we strongly recommend using a unique key for each peer.

---

**Step 4** Specify the pre-shared key at the remote IPSec peer.

If the remote peer is a PIX Firewall, use the same command as shown in Step 3.



**Note**

---

The pre-shared key should be configured at both the PIX Firewall and its peer, otherwise the policy cannot be used. Configure a pre-shared key associated with a given security gateway to be distinct from a wildcard, pre-shared key (pre-shared key plus a netmask of 0.0.0.0) used to identify and authenticate the remote VPN clients.

---

## Using Certification Authorities

This section provides background information about certification authorities (CAs) and describes how to configure the PIX Firewall to work with a CA. It includes the following topics:

- [CA Overview, page 6-8](#)
- [Public Key Cryptography, page 6-8](#)
- [Certificates Provide Scalability, page 6-8](#)
- [Supported CA Servers, page 6-9](#)
- [Configuring the PIX Firewall to Use Certificates, page 6-9](#)

## CA Overview

Certification authorities (CAs) are responsible for managing certificate requests and issuing digital certificates. A digital certificate contains information that identifies a user or device, such as a name, serial number, company, department, or IP address. A digital certificate also contains a copy of the entity's public key. A CA can be a trusted third party, such as VeriSign, or a private (in-house) CA that you establish within your organization.

## Public Key Cryptography

Digital signatures, enabled by public key cryptography, provide a means to digitally authenticate devices and individual users. In public key cryptography, such as the RSA encryption system, each user has a key-pair containing both a public and a private key. The keys act as complements, and anything encrypted with one of the keys can be decrypted with the other. In simple terms, a signature is formed when data is encrypted with a user's private key. The receiver verifies the signature by decrypting the message with the sender's public key.

The fact that the message could be decrypted using the sender's public key means that the holder of the private key created the message. This process relies on the receiver having a copy of the sender's public key and knowing with a high degree of certainty that it really does belong to the sender, and not to someone pretending to be the sender.

To validate the CA's signature, the receiver must know the CA's public key. Normally this is handled out-of-band or through an operation done at installation. For instance, most web browsers are configured with the root certificates of several CAs by default. The IKE, a key component of IPSec, can use digital signatures to authenticate peer devices before setting up security associations.

## Certificates Provide Scalability

Without digital certificates, each IPSec peer must be manually configured for every peer with which it communicates. Without certificates, every new peer added to the network requires a configuration change on every other peer it securely communicates with. However, when using digital certificates, each peer is enrolled with a CA. When two peers wish to communicate, they exchange certificates and digitally sign data to authenticate each other.

When a new peer is added to the network, one simply enrolls that peer with a CA, and none of the other peers need modification. When the new peer attempts an IPSec connection, certificates are automatically exchanged and the peer can be authenticated.

With a CA, a peer authenticates itself to the remote peer by sending a certificate to the remote peer and performing some public key cryptography. Each peer sends its own unique certificate which was issued and validated by the CA. This process works because each peer's certificate encapsulates the peer's public key, each certificate is authenticated by the CA, and all participating peers recognize the CA as an authenticating authority. This is called IKE with an RSA signature.

The peer can continue sending its own certificate for multiple IPSec sessions, and to multiple IPSec peers, until the certificate expires. When its certificate expires, the peer administrator must obtain a new one from the CA.

CAs can also revoke certificates for peers that will no longer participate in IPSec. Revoked certificates are not recognized as valid by other peers. Revoked certificates are listed in a certificate revocation list (CRL), which each peer may check before accepting another peer's certificate.

Some CAs have a registration authority (RA) as part of their implementation. An RA is essentially a server that acts as a proxy for the CA so that CA functions can continue when the CA is off line.

## Supported CA Servers

Currently, the PIX Firewall supports the following CA servers:

- VeriSign support is provided through the VeriSign Private Certificate Services (PCS) and the OnSite service, which lets you establish an in-house CA system for issuing digital certificates.
- Entrust, Entrust VPN Connector, version 4.1 (build 4.1.0.337) or higher. The Entrust CA server is an in-house CA server solution.
- Baltimore Technologies, UniCERT Certificate Management System, version 3.1.2 or higher. The Baltimore CA server is an in-house CA server solution.
- Microsoft Windows 2003 Server, Microsoft Windows 2000, specifically the Windows 2000 Advanced Server, version 5.00.2195 or higher. The Windows 2000 CA server is an in-house CA server solution.

**Note**

---

The Microsoft CA must be a standalone root CA, not subordinated, or it will be rejected and a syslog CRYPTO\_PKI: WARNING message will be entered. Example: CRYPTO\_PKI: WARNING: A certificate chain could not be constructed while selecting certificate status.

---

## Configuring the PIX Firewall to Use Certificates

For site-to-site VPNs, you must perform this series of steps for each PIX Firewall. For remote access VPNs, perform these steps for each PIX Firewall and each remote access VPN client.

**Note**

---

You need to have a CA available to your network before you configure CA. The CA should support Cisco's PKI protocol, the simple certificate enrollment protocol.

---

When certificates are revoked, they are added to a certificate revocation list (CRL). When you implement authentication using certificates, you can choose to use CRLs or not. Using CRLs lets you easily revoke certificates before they expire, but the CRL is generally only maintained by the CA or its authorized registration authority (RA). If you are using CRLs and the connection to the CA or RA is not available when authentication is requested, the authentication request will fail.

**Note**

---

Be sure that the PIX Firewall clock is set to GMT, month, day, and year before configuring the CA. Otherwise, the CA may reject or allow certificates based on an incorrect timestamp. Cisco's PKI protocol uses the clock to make sure that a CRL is not expired. The lifetime of a certificate and CRL is checked in GMT time. If you are using IPSec with certificates, set the PIX Firewall clock to GMT to ensure that CRL checking works correctly.

---

Follow these steps to enable your PIX Firewall to interoperate with a CA and obtain your PIX Firewall certificate(s):

**Step 1** Configure the PIX Firewall host name:

```
hostname newname
```

For example:

```
hostname mypixfirewall
```

In this example, “mypixfirewall” is the name of a unique host in the domain.

**Step 2** Configure the PIX Firewall domain name:

```
domain-name name
```

For example:

```
domain-name example.com
```

**Step 3** Generate the PIX Firewall RSA key pair(s):

```
ca generate rsa key key_modulus_size
```

For example:

```
ca generate rsa key 512
```

In this example, one general purpose RSA key pair is to be generated. The other option is to generate two special-purpose keys. The selected size of the key modulus is 512.

**Step 4** (Optional) View your RSA key pair(s):

```
show ca mypubkey rsa
```

The following is sample output from the **show ca mypubkey rsa** command:

```
show ca mypubkey rsa
```

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

```
Key name: mypixfirewall.example.com
```

```
Usage: General Purpose 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
```

**Step 5** Declare a CA:

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

For example:

```
ca identity myca.example.com 209.165.202.130
```

In this example, 209.165.202.130 is the IP address of the CA. The CA name is myca.example.com.



**Note** The CA may require a particular name for you to use, such as its domain name. When using VeriSign as your CA, VeriSign assigns the CA name you are to use in your CA configuration.

**Step 6** Configure the parameters of communication between the PIX Firewall and the CA:

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

For example:

```
ca configure myca.example.com ca 1 20 crloptional
```

If the PIX Firewall does not receive a certificate from the CA within 1 minute (the default) of sending a certificate request, it will resend the certificate request. The PIX Firewall will continue sending a certificate request every 1 minute until a certificate is received or until 20 requests have been sent. With the keyword **crloptional** included within the command statement, other peer's certificates can still be accepted by your PIX Firewall even if the CRL is not accessible to your PIX Firewall.

**Step 7** Authenticate the CA by obtaining its public key and its certificate:

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

For example:

```
ca authenticate myca.example.com 0123 4567 89AB CDEF 0123
```

The fingerprint (0123 4567 89AB CDEF 0123 in the example) is optional and is used to authenticate the CA's public key within its certificate. The PIX Firewall will discard the CA certificate if the fingerprint that you included in the command statement is not equal to the fingerprint within the CA's certificate.

You also have the option to manually authenticate the public key by simply comparing the two fingerprints after you receive the CA's certificate rather than entering it within the command statement.



---

**Note** Depending on the CA you are using, you may need to ask your local CA administrator for this fingerprint.

---

**Step 8** Request signed certificates from your CA for all of your PIX Firewall's RSA key pairs. Before entering this command, contact your CA administrator because they must authenticate your PIX Firewall manually before granting its certificate(s).

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

For example:

```
ca enroll myca.example.com mypassword1234567 serial ipaddress
```

The keyword `mypassword1234567` in the example is a password, which is not saved with the configuration. The options "serial" and "ipaddress" are included, which indicates the PIX Firewall unit's serial number and IP address will be included in the signed certificate.



---

**Note** The password is required in the event your certificate needs to be revoked, so it is crucial that you remember this password. Note it and store it in a safe place.

---

The **ca enroll** command requests as many certificates as there are RSA key pairs. You will only need to perform this command once, even if you have special usage RSA key pairs.



---

**Note** If your PIX Firewall reboots after you issued the **ca enroll** command but before you received the certificate(s), reissue the command and notify the CA administrator.

---

**Step 9** Verify that the enrollment process was successful using the **show ca certificate** command:

```
show ca certificate
```

The following is sample output from the **show ca certificate** command including a PIX Firewall general purpose certificate and the RA and CA public-key certificates:

```
Subject Name
  Name: mypixfirewall.example.com
IP Address: 192.150.50.110
  Status: Available
  Certificate Serial Number: 36f97573
  Key Usage: General Purpose

RA Signature Certificate
  Status: Available
  Certificate Serial Number: 36f972f4
  Key Usage: Signature

CA Certificate
  Status: Available
  Certificate Serial Number: 36f972e5
  Key Usage: Not Set

RA KeyEncipher Certificate
  Status: Available
  Certificate Serial Number: 36f972f3
  Key Usage: Encryption
```

**Step 10** Save the configuration:

```
ca save all
write memory
```

## Verifying the Distinguished Name of a Certificate

PIX Firewall Version 6.3 lets you specify the distinguished name (DN) of the certificate used to establish a VPN tunnel. We recommend enabling this feature to prevent a possible “man-in-the-middle” attack.

To verify the DN of the certificate received by your PIX Firewall, enter the following command:

```
ca verifycertdn x500 string
```



### Note

Every attribute must match exactly to verify the certificate received and to establish a VPN tunnel.

For example, a PIX Firewall might have the following certificate:

```
Certificate
  Status: Available
  Certificate Serial Number: 4ebdbd400000000000a2
  Key Usage: General Purpose
  Subject Name:
    CN = myvpn01.myorg.com
    OU = myou
    O = myorg
    ST = CA
    C = US
    UNSTRUCTURED NAME = myvpn01.myorg.com
```

```
Validity Date:
  start date: 23:48:00 UTC Feb 18 2003
  end   date: 23:58:00 UTC Feb 18 2004
-----
```

To establish a VPN tunnel with this server, enter the following command on the PIX Firewall that will receive this certificate:

```
ca verifycertdn cn*myvpn, ou=myou, o=myorg, st=ca, c=US
```

This command causes the receiving PIX Firewall to accept certificates with any DN having the following attributes:

- Common name (CN) contains the string *myvpn*
- Organizational unit (OU) equals *myou*
- Organization (O) equals *myorg*
- State (ST) equals *CA*
- Country (C) equals *US*

You could be more restrictive by identifying a specific common name, or less restrictive by omitting the CN attribute altogether.

You can use an asterisk (\*) to match an attribute containing the string following the asterisk. Use an exclamation mark (!) to match an attribute that does not contain the characters following the exclamation mark.

## Configuring IPSec

This section provides background information about IPSec and describes the procedures required to configure the PIX Firewall when using IPSec to implement a VPN. It contains the following topics:

- [IPSec Overview, page 6-13](#)
- [Transform Sets, page 6-14](#)
- [Crypto Maps, page 6-14](#)
- [Applying Crypto Maps to Interfaces, page 6-16](#)
- [Access Lists, page 6-16](#)
- [IPSec SA Lifetimes, page 6-18](#)
- [Basic IPSec Configuration, page 6-19](#)
- [Diffie-Hellman Group 5, page 6-21](#)
- [Using Dynamic Crypto Maps, page 6-22](#)
- [Site-to-Site Redundancy, page 6-24](#)

## IPSec Overview

IPSec tunnels are sets of security associations that are established between two remote IPSec peers. The security associations define which protocols and algorithms should be applied to sensitive packets, and also specify the keying material to be used by the two peers. IPSec SAs are used during the actual transmission of user traffic. SAs are unidirectional and are established separately for different security protocols (AH and/or ESP).

You can establish IPSec SAs in two ways:

- **Manual SAs with Pre-Shared Keys** —The use of manual IPSec SAs requires a prior agreement between administrators of the PIX Firewall and the IPSec peer. There is no negotiation of SAs, so the configuration information in both systems should be the same for traffic to be processed successfully by IPSec.
- **IKE-Established SAs**—When IKE is used to establish IPSec SAs, the peers can negotiate the settings they will use for the new security associations. This means that you can specify lists (such as lists of acceptable transforms) within the crypto map entry.

The PIX Firewall can simultaneously support manual and IKE-established security associations.

## Transform Sets

A transform set represents a certain combination of security protocols and algorithms. During the IPSec security association negotiation, the peers agree to use a particular transform set for protecting a particular data flow.

You can specify multiple transform sets, and then specify one or more of these transform sets in a crypto map entry. The transform set defined in the crypto map entry will be used in the IPSec security association negotiation to protect the data flows specified by that crypto map entry's access list.

During IPSec security association negotiations with IKE, the peers search for a transform set that is the same at both peers. When such a transform set is found, it is selected and will be applied to the protected traffic as part of both peers' IPSec security associations. With manually established security associations, there is no negotiation with the peer, so both sides have to specify the same transform set.

If you change a transform set definition, 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, clear all or part of the security association database by using the **clear [crypto] ipsec sa** command. See "[Clearing SAs](#)" for further information.

## Crypto Maps

Crypto maps specify IPSec policy. Crypto map entries created for IPSec pull together the various parts used to set up IPSec security associations, including the following:

- Which traffic should be protected by IPSec (per a crypto access list)
- Where IPSec-protected traffic should be sent (who the peer is)
- The local address to be used for the IPSec traffic (See "[Applying Crypto Maps to Interfaces](#)" for more details.)
- What IPSec security should be applied to this traffic (selecting from a list of one or more transform sets)
- Whether security associations are manually established or are established via IKE
- Other parameters that might be necessary to define an IPSec SA

Crypto map entries with the same crypto map name (but different map sequence numbers) are grouped into a crypto map set. Later, you will apply these crypto map sets to interfaces; then, all IP traffic passing through the interface is evaluated against the applied crypto map set. If a crypto map entry sees outbound IP traffic that should be protected and the crypto map specifies the use of IKE, a security association is negotiated with the peer according to the parameters included in the crypto map entry; otherwise, if the crypto map entry specifies the use of manual security associations, a security association should have already been established via configuration. (If a dynamic crypto map entry sees outbound traffic that should be protected and no security association exists, the packet is dropped.)

The policy described in the crypto map entries is used during the negotiation of security associations. If the local PIX Firewall initiates the negotiation, it will use the policy specified in the static crypto map entries to create the offer to be sent to the specified peer. If the peer initiates the negotiation, the PIX Firewall will check the policy from the static crypto map entries, as well as any referenced dynamic crypto map entries to decide whether to accept or reject the peer's request (offer).

For IPSec to succeed between two peers, both peers' crypto map entries have to contain compatible configuration statements.

When two peers try to establish a security association, they should each have at least one crypto map entry that is compatible with one of the other peer's crypto map entries. For two crypto map entries to be compatible, they should, at a minimum, meet the following criteria:

- The crypto map entries contain compatible crypto access lists (for example, mirror image access lists). In the case where the responding peer is using dynamic crypto maps, the entries in the PIX Firewall crypto access list must be "permitted" by the peer's crypto access list.
- The crypto map entries each identify the other peer (unless the responding peer is using dynamic crypto maps).
- The crypto map entries have at least one transform set in common.

You can apply only one crypto map set to a single interface. The crypto map set can include a combination of IPSec/IKE and IPSec/manual entries.

If you create more than one crypto map entry for a given interface, use the seq-num of each map entry to rank the map entries: the lower the seq-num, the higher the priority. At the interface that has the crypto map set, traffic is evaluated against higher priority map entries first.

Create multiple crypto map entries for a given PIX Firewall interface, if any of the following conditions exist:

- If different data flows are to be handled by separate peers.
- If you want to apply different IPSec security to different types of traffic (to the same or separate peers); for example, if you want traffic between one set of subnets to be authenticated, and traffic between another set of subnets to be both authenticated and encrypted. In this case, the different types of traffic should have been defined in two separate access lists, and you create a separate crypto map entry for each crypto access list.
- If you are configuring manual SAs to establish a particular set of IPSec security associations, and want to specify multiple access list entries, create separate access lists (one per permit entry) and specify a separate crypto map entry for each access list.

## Applying Crypto Maps to Interfaces

You must apply a crypto map set to each interface through which IPsec traffic will flow. The PIX Firewall supports IPsec on all of its interfaces. Applying the crypto map set to an interface instructs the PIX Firewall to evaluate all the interface's traffic against the crypto map set and to use the specified policy during connection or security association negotiation on behalf of traffic to be protected by crypto IPsec.

Binding a crypto map to an interface will also initialize the run-time data structures, such as the security association database and the security policy database. If the crypto map is modified in any way, reapplying the crypto map to the interface will resynchronize the various run-time data structures with the crypto map configuration. In addition, any existing connections will be torn down and will be reestablished after the new crypto map is triggered.

## Access Lists

By default, IPsec and all packets that traverse the PIX Firewall are subjected to blocking as specified by access lists. To enable IPsec packets to traverse the PIX Firewall, ensure that you have statements in access lists that permit the packets. Optionally, the **sysopt connection permit-ipsec** command can be configured to enable IPsec packets to bypass access list blocking.



### Note

The **sysopt connection permit-ipsec** command enables packets that have been processed by IPsec to bypass access list checks.

IPsec packets that are destined to an IPsec tunnel are selected by the crypto map access list bound to the outgoing interface. IPsec packets that arrive from an IPsec tunnel are authenticated/deciphered by IPsec, and are subjected to the proxy identity match of the tunnel.

Crypto access lists are used to define which IP traffic will be protected by crypto and which traffic will not be protected by crypto. For example, access lists can be created to protect all IP traffic between Subnet A and Subnet Y or between Host A and Host B. (These access lists are similar to access lists used with the **access-group** command. With the **access-group** command, the access list determines which traffic to forward or block at an interface.)

The access lists themselves are not specific to IPsec. It is the crypto map entry referencing the specific access list that defines whether IPsec processing is applied to the traffic matching a permit in the access list.

Crypto access lists associated with IPsec crypto map entries have four primary functions:

- Select outbound traffic to be protected by IPsec (permit = protect).
- Indicate the data flow to be protected by the new security associations (specified by a single permit entry) when initiating negotiations for IPsec security associations.
- Process inbound traffic to filter out and discard traffic that should have been protected by IPsec.
- Determine whether or not to accept requests for IPsec security associations on behalf of the requested data flows when processing IKE negotiation from the peer. (Negotiation is only done for **ipsec-isakmp crypto map** entries.) For the peer's request to be accepted during negotiation, the peer should specify a data flow that is "permitted" by a crypto access list associated with an **ipsec-isakmp crypto map** command entry.

If you want certain traffic to receive one combination of IPSec protection (for example, authentication only) and other traffic to receive a different combination of IPSec protection (for example, both authentication and encryption), you must create two different crypto access lists to define the two different types of traffic. These different access lists are then used in different crypto map entries which specify different IPSec policies.

Using the **permit** keyword causes all IP traffic that matches the specified conditions to be protected by crypto, using the policy described by the corresponding crypto map entry. Using the **deny** keyword prevents traffic from being protected by crypto IPSec in the context of that particular crypto map entry. (In other words, it does not allow the policy as specified in this crypto map entry to be applied to this traffic.) If this traffic is denied in all the crypto map entries for that interface, the traffic is not protected by crypto IPSec.

The crypto access list you define will be applied to an interface after you define the corresponding crypto map entry and apply the crypto map set to the interface. Different access lists should be used in different entries of the same crypto map set. However, both inbound and outbound traffic will be evaluated against the same “outbound” IPSec access list.

Therefore, the access list’s criteria are applied in the forward direction to traffic exiting your PIX Firewall, and the reverse direction to traffic entering your PIX Firewall. In [Figure 6-1](#), IPSec protection is applied to traffic between Host 10.0.0.1 and Host 10.2.2.2 as the data exits PIX Firewall A’s outside interface toward Host 10.2.2.2. For traffic from Host 10.0.0.1 to Host 10.2.2.2, the access list entry on PIX Firewall A is evaluated as follows:

source = host 10.0.0.1

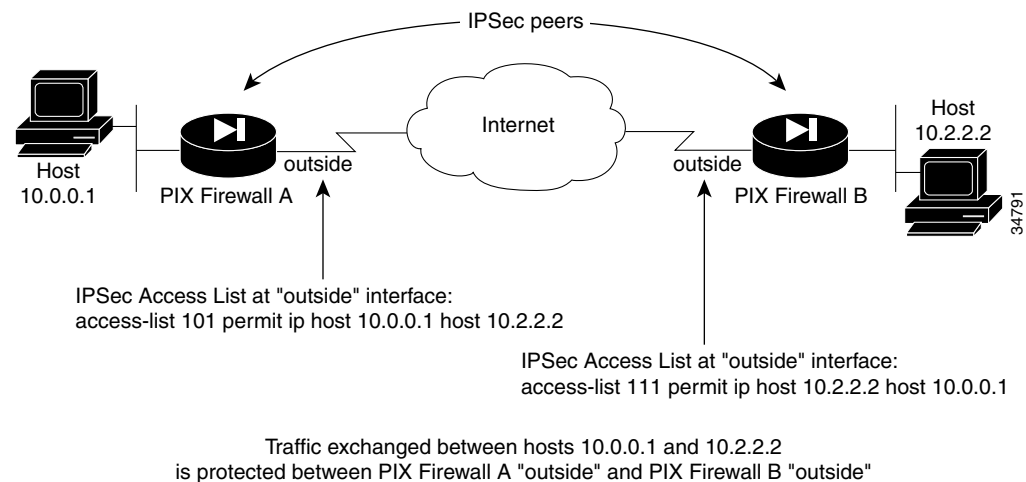
dest = host 10.2.2.2

For traffic from Host 10.2.2.2 to Host 10.0.0.1, that same access list entry on PIX Firewall A is evaluated as follows:

source = host 10.2.2.2

dest = host 10.0.0.1

**Figure 6-1 How Crypto Access Lists Are Applied for Processing IPSec**



If you configure multiple statements for a given crypto access list that is used for IPSec, in general the first permit statement that is matched will be the statement used to determine the scope of the IPSec security association. That is, the IPSec security association will be set up to protect traffic that meets the

criteria of the matched statement only. Later, if traffic matches a different permit statement of the crypto access list, a new, separate IPsec security association will be negotiated to protect traffic matching the newly matched access list statement.

Any unprotected inbound traffic that matches a permit entry in the crypto access list for a crypto map entry flagged as IPsec will be dropped because this traffic was expected to be protected by IPsec.

Access lists for crypto map entries tagged as `ipsec-manual` are restricted to a single permit entry and subsequent entries are ignored. In other words, the security associations established by that particular crypto map entry are only for a single data flow. To support multiple manually established security associations for different kinds of traffic, define multiple crypto access lists, and apply each one to a separate **ipsec-manual crypto map** command entry. Each access list should include one permit statement defining which traffic to protect.

**Note**


---

If you clear or delete the last element from an access list, the crypto map references to the destroyed access list are also removed.

---

If you modify an access list that is currently referenced by one or more crypto map entries, the run-time security association database will need to be re initialized using the **crypto map interface** command. See the **crypto map** command page for more information.

We recommend that for every crypto access list specified for a static crypto map entry that you define at the local peer, you define a “mirror image” crypto access list at the remote peer. This ensures that traffic that has IPsec protection applied locally can be processed correctly at the remote peer. (The crypto map entries themselves should also support common transforms and refer to the other system as a peer.)

**Note**


---

Every static crypto map must define an access list and an IPsec peer. If either is missing, the crypto map is considered incomplete and any traffic that has not already been matched to an earlier, complete crypto map is dropped. Use the **show conf** command to ensure that every crypto map is complete. To fix an incomplete crypto map, remove the crypto map, add the missing entries, and reapply it.

---

When you create crypto access lists, using the **any** keyword could cause problems. We discourage the use of the **any** keyword to specify source or destination addresses.

The **permit any any** command statement is strongly discouraged, as this will cause all outbound traffic to be protected (and all protected traffic sent to the peer specified in the corresponding crypto map entry) and will require protection for all inbound traffic. Then, all inbound packets that lack IPsec protection will be silently dropped.

You must be sure that you define which packets to protect. If you use the **any** keyword in a **permit** command statement, preface that statement with a series of **deny** command statements to filter out any traffic (that would otherwise fall within that **permit** command statement) that you do not want to be protected.

## IPsec SA Lifetimes

You can change the global lifetime values that are used when negotiating new IPsec security associations. (These global lifetime values can be overridden for a particular crypto map entry.)

These lifetimes only apply to security associations established via IKE. Manually established security associations do not expire.

There are two lifetimes: a “timed” lifetime and a “traffic-volume” lifetime. A security association expires after the respective lifetime is reached and negotiations will be initiated for a new one. The default lifetimes are 28,800 seconds (eight hours) and 4,608,000 kilobytes (10 megabytes per second for one hour).

If you change a global lifetime, the new lifetime value will not be applied to currently existing security associations, but will be used in the negotiation of subsequently established security associations. If you wish to use the new values immediately, you can clear all or part of the security association database. See the **clear [crypto] ipsec sa** command for more information within the **crypto ipsec** command page of *Cisco PIX Firewall Command Reference*.

IPSec security associations use one or more 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 it will specify its global lifetime values in the request to the peer; it will use this value as the lifetime of the new security associations. When the PIX Firewall receives a negotiation request from the peer, it will use the smaller of either the lifetime value proposed by the peer or the locally configured lifetime value as the lifetime of the new security associations.

The security association and the corresponding keys expire after a configurable interval of time or after forwarding a configurable volume of traffic.

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.

## Basic IPSec Configuration

The following steps cover basic IPSec configuration where the IPSec security associations are established with IKE and static crypto maps are used. For information about configuring IPSec for specific implementations, see the following chapters:

- [Chapter 7, “Site-to-Site VPN Configuration Examples.”](#)
- [Chapter 8, “Managing VPN Remote Access.”](#)

In general, to configure the PIX Firewall for using IPSec, perform the following steps:

---

**Step 1** Create an access list to define the traffic to protect:

```
access-list access-list-name {deny | permit} ip source source-netmask destination
destination-netmask
```

For example:

```
access-list 101 permit ip 10.0.0.0 255.255.255.0 10.1.1.0 255.255.255.0
```

In this example, the **permit** keyword causes all traffic that matches the specified conditions to be protected by crypto.

- Step 2** Configure a transform set that defines how the traffic will be protected. You can configure multiple transform sets, and then specify one or more of these transform sets in a crypto map entry (Step 3d).

```
crypto ipsec transform-set transform-set-name transform1 [transform2, transform3]
```

For example:

```
crypto ipsec transform-set myset1 esp-des esp-sha-hmac
crypto ipsec transform-set myset2 ah-sha-hmac esp-3des esp-sha-hmac
crypto ipsec transform-set aes_set ah-md5-hmac esp-aes-256
```




---

**Note** PIX Firewall version 6.3 introduces support for AES, which provides for encryption keys of 128, 192, and 256 bits.

---

In this example, “myset1” and “myset2” are the names of the transform sets. “myset1” has two transforms defined, while “myset2” has three transforms defined.

- Step 3** Create a crypto map entry by performing the following steps:

- a. Create a crypto map entry in IPSec ISAKMP mode:

```
crypto map map-name seq-num ipsec-isakmp
```

For example:

```
crypto map mymap 10 ipsec-isakmp
```

In this example, “mymap” is the name of the crypto map set. The map set’s sequence number is 10, which is used to rank multiple entries within one crypto map set. The lower the sequence number, the higher the priority.

- b. Assign an access list to a crypto map entry:

```
crypto map map-name seq-num match address access-list-name
```

For example:

```
crypto map mymap 10 match address 101
```

In this example, access list 101 is assigned to crypto map “mymap.”

- c. Specify the peer to which the IPSec protected traffic can be forwarded:

```
crypto map map-name seq-num set peer ip-address
```

For example:

```
crypto map mymap 10 set peer 192.168.1.100
```

The security association will be set up with the peer having an IP address of 192.168.1.100. Specify multiple peers by repeating this command.

- d. Specify which transform sets are allowed for this crypto map entry. List multiple transform sets in order of priority (highest priority first). You can specify up to six transform sets.

```
crypto map map-name seq-num set transform-set transform-set-name1
[transform-set-name2, ...transform-set-name6]
```

For example:

```
crypto map mymap 10 set transform-set myset1 myset2
```

In this example, when traffic matches access list 101, the security association can use either “myset1” (first priority) or “myset2” (second priority) depending on which transform set matches the peer’s transform set.

- e. (Optional) Specify security association lifetime for the crypto map entry, if you want the security associations for this entry to be negotiated using different IPSec security association lifetimes other than the global lifetimes.

```
crypto map map-name seq-num set security-association lifetime {seconds seconds |
kilobytes kilobytes}
```

For example:

```
crypto map mymap 10 set security-association lifetime seconds 2700
```

This example shortens the timed lifetime for the crypto map “mymap 10” to 2700 seconds (45 minutes). The traffic volume lifetime is not changed.

- f. (Optional) Specify that IPSec should ask for perfect forward secrecy (PFS) when requesting new security associations for this crypto map entry, or should require PFS in requests received from the peer:

```
crypto map map-name seq-num set pfs [group1 | group2 | group5]
```




---

**Note** Support for Diffie-Hellman group 5 is introduced with PIX Firewall version 6.3.

---

For example:

```
crypto map mymap 10 set pfs group2
```

This example specifies that PFS should be used whenever a new security association is negotiated for the crypto map “mymap 10.” The 1024-bit Diffie-Hellman prime modulus group will be used when a new security association is negotiated using the Diffie-Hellman exchange.

- Step 4** Apply a crypto map set to an interface on which the IPSec traffic will be evaluated:

```
crypto map map-name interface interface-name
```

For example:

```
crypto map mymap interface outside
```

In this example, the PIX Firewall will evaluate the traffic going through the outside interface against the crypto map “mymap” to determine whether it needs to be protected.

- Step 5** Specify that IPSec traffic be implicitly trusted (permitted):

```
sysopt connection permit-ipsec
```

---

## Diffie-Hellman Group 5

Diffie-Hellman is a public key operation that provides a method for two IPSec peers to agree on a key to use. To perform the Diffie-Hellman operation, both sides must agree to use a number or group for the mathematical calculation. Versions of PIX Firewall prior to Version 6.3 support group 1 (768 bits) and group 2 (1024 bits). PIX Firewall Version 6.3 introduces support for Group 5 (1536 bits), which provides higher security for the Diffie-Hellman operation. In version 6.3, PIX Firewall also supports AES (Advance Encryption Standard) which provides cryptographic keys of 256 bits and which requires the use of Diffie-Hellman Group 5 keys.

# Using Dynamic Crypto Maps

Dynamic crypto maps, used with IKE, can ease IPSec configuration and are recommended for use in networks where the peers are not always predetermined. You use dynamic crypto maps for VPN clients (such as mobile users) and routers that obtain dynamically assigned IP addresses. For an example of using dynamic crypto maps in a remote access VPN configuration, see [Chapter 8, “Managing VPN Remote Access.”](#)

**Note**

Use care when using the **any** keyword in **permit** command entries in dynamic crypto maps. If it is possible for the traffic covered by such a **permit** command entry to include multicast or broadcast traffic, the access list should include **deny** command entries for the appropriate address range. Access lists should also include **deny** command entries for network and subnet broadcast traffic, and for any other traffic that should not be IPSec protected.

Dynamic crypto maps can only be used for negotiating SAs with remote peers that initiate the connection. They cannot be used for initiating connections to a remote peer. With a dynamic crypto map entry, if outbound traffic matches a permit statement in an access list and the corresponding security association is not yet established, the PIX Firewall will drop the traffic.

A dynamic crypto map entry is essentially a crypto map entry without all the parameters configured. It acts as a policy template where the missing parameters are later dynamically configured (as the result of an IPSec negotiation) to match a peer's requirements. This allows peers to exchange IPSec traffic with the PIX Firewall even if the PIX Firewall does not have a crypto map entry specifically configured to meet all the peer's requirements.

**Note**

Only the **transform-set** parameter is required to be configured within each dynamic crypto map entry.

A dynamic crypto map set is included by reference as part of a crypto map set. Any crypto map entries that reference dynamic crypto map sets should be the lowest priority crypto map entries in the crypto map set (that is, have the highest sequence numbers) so that the other crypto map entries are evaluated first; that way, the dynamic crypto map set is examined only when the other (static) map entries are not successfully matched.

If the 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 the corresponding security associations expire), the temporary crypto map entry is then removed.

Dynamic crypto map entries, like regular static crypto map entries, are grouped into sets. A set is a group of dynamic crypto map entries all with the same dynamic-map-name but each with a different dynamic-seq-num. If this is configured, the data flow identity proposed by the IPSec peer should fall within a **permit** statement for this crypto access list. If this is not configured, the PIX Firewall will accept any data flow identity proposed by the peer.

You can add one or more dynamic crypto map sets into a crypto map set via crypto map entries that reference the dynamic crypto map sets. You should set the crypto map entries referencing dynamic maps to be the lowest priority entries in a crypto map set (that is, use the highest sequence numbers).

**Note**

Use care when using the **any** keyword in **permit** entries in dynamic crypto maps. If it is possible for the traffic covered by such a **permit** entry to include multicast or broadcast traffic, the access list should include deny entries for the appropriate address range. Access lists should also include **deny** entries for network and subnet broadcast traffic, and for any other traffic that should not be IPSec protected.

The procedure for using a crypto dynamic map entry is the same as the basic configuration described in “[Basic IPSec Configuration](#),” except that instead of creating a static crypto map entry, you create a crypto dynamic map entry. You can also combine static and dynamic map entries within a single crypto map set.

Create a crypto dynamic map entry by performing the following steps:

**Step 1** Assign an access list to a dynamic crypto map entry:

```
crypto dynamic-map dynamic-map-name dynamic-seq-num match address access-list-name
```

This determines which traffic should be protected and not protected.

For example:

```
crypto dynamic-map dyn1 10 match address 101
```

In this example, access list 101 is assigned to dynamic crypto map “dyn1.” The map’s sequence number is 10.

**Step 2** Specify which transform sets are allowed for this dynamic crypto map entry. List multiple transform sets in order of priority (highest priority first).

```
crypto dynamic-map dynamic-map-name dynamic-seq-num set transform-set transform-set-name1,  
[transform-set-name2, ...transform-set-name9]
```

For example:

```
crypto dynamic-map dyn 10 set transform-set myset1 myset2
```

In this example, when traffic matches access list 101, the security association can use either “myset1” (first priority) or “myset2” (second priority) depending on which transform set matches the peer’s transform sets.

**Step 3** Specify security association lifetime for the crypto dynamic map entry, if you want the security associations for this entry to be negotiated using different IPSec security association lifetimes other than the global lifetimes:

```
crypto dynamic-map dynamic-map-name dynamic-seq-num set security-association lifetime  
{seconds seconds | kilobytes kilobytes}
```

For example:

```
crypto dynamic-map dyn1 10 set security-association lifetime 2700
```

This example shortens the timed lifetime for dynamic crypto map “dyn1 10” to 2700 seconds (45 minutes). The time volume lifetime is not changed.

- Step 4** Specify that IPsec should ask for PFS when requesting new security associations for this dynamic crypto map entry, or should demand PFS in requests received from the peer:

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

For example:

```
crypto dynamic-map dyn1 10 set pfs group1
```

- Step 5** Add the dynamic crypto map set into a static crypto map set.

Be sure to set the crypto map entries referencing dynamic maps to be the lowest priority entries (highest sequence numbers) in a crypto map set.

```
crypto map map-name seq-num ipsec-isakmp dynamic dynamic-map-name
```

For example:

```
crypto map mymap 200 ipsec-isakmp dynamic dyn1
```

---

## Site-to-Site Redundancy

You can define multiple peers by using crypto maps to allow for redundancy. This configuration is also most useful for site-to-site VPNs. If one peer fails, there will still be a protected path. The peer that packets are actually sent to is determined by the last peer that the PIX Firewall heard from (received either traffic or a negotiation request from) for a given data flow. If the attempt fails with the first peer, IKE tries the next peer on the crypto map list.

## Using NAT Traversal

Network Address Translation (NAT) and Port Address Translation (PAT) are implemented in many networks where IPsec is also used, but the a number of incompatibilities that prevent IPsec packets from successfully traversing a NAT device.

PIX Firewall Version 6.3 provides a feature called “Nat Traversal,” as described by Version 2 and Version 3 of the draft IETF standard, UDP Encapsulation of IPsec Packets,” which is available at the following URL:

<http://www.ietf.org/html.charters/ipsec-charter.html>

NAT Traversal allows ESP packets to pass through one or more NAT devices. This feature is disabled by default.



### Note

NAT Traversal is supported for both dynamic and static crypto maps.

To enable NAT traversal, enter the following command:

```
isakmp nat-traversal [natkeepalive]
```

Valid values for *natkeepalive* are 10 to 3600 seconds; the default is 20 seconds.

# Manual Configuration of SAs

When you cannot use IKE to establish SAs between your PIX Firewall and a remote IPSec peer, you can manually configure the SAs. This is only practical with a limited number of IPSec peers having known IP addresses (or DNS host names), so this method of configuration is most practical for site-to-site VPNs.

Manually configuring SAs is very similar to the basic configuration described in “[Configuring IPSec](#).” The following are the main differences:

- The crypto map is configured using the **ipsec-manual** keyword, as in the following example:

```
crypto map map-name seq-num ipsec-manual
```

- SA lifetimes and perfect forward secrecy (PFS) are not configurable
- You manually configure the session keys on both IPSec peers

When you manually configure SAs, you lose the benefits of enhanced security and scalability that IKE can provide. Manually configure each pair of IPSec peers that communicate securely, and session keys do not change unless you manually reconfigure the SAs.



## Note

Manual configuration of SAs is not supported on the PIX 501 because of the restriction in the number of ISAKMP peers allowed on that platform.

To manually configure SAs, perform the following steps:

- Step 1** Create an access list to define the traffic to protect:

```
access-list access-list-name {deny | permit} ip source source-netmask destination
destination-netmask
```

For example:

```
access-list 101 permit ip 10.0.0.0 255.255.255.0 10.1.1.0 255.255.255.0
```

In this example, the **permit** keyword causes all traffic that matches the specified conditions to be protected by crypto.

- Step 2** Configure a transform set that defines how the traffic will be protected. You can configure multiple transform sets, and then specify one or more of these transform sets in a crypto map entry (Step 4d).

```
crypto ipsec transform-set transform-set-name transform1 [transform2, transform3]
```

For example:

```
crypto ipsec transform-set myset1 esp-des esp-sha-hmac
crypto ipsec transform-set myset2 ah-sha-hmac esp-3des esp-sha-hmac
```

In this example, “myset1” and “myset2” are the names of the transform sets. “myset1” has two transforms defined, while “myset2” has three transforms defined.

**Step 3** Create a crypto map entry by performing the following steps:

- a. Create a crypto map entry in IPSec manual configuration mode:

```
crypto map map-name seq-num ipsec-manual
```

For example:

```
crypto map mymap 10 ipsec-manual
```

In this example, “mymap” is the name of the crypto map set. The map set’s sequence number is 10, which is used to rank multiple entries within one crypto map set. The lower the sequence number, the higher the priority.

- b. Assign an access list to a crypto map entry:

```
crypto map map-name seq-num match address access-list-name
```

For example:

```
crypto map mymap 10 match address 101
```

In this example, access list 101 is assigned to crypto map “mymap.”

- c. Specify the peer to which the IPSec protected traffic can be forwarded:

```
crypto map map-name seq-num set peer ip-address
```

For example:

```
crypto map mymap 10 set peer 192.168.1.100
```

The security association will be set up with the peer having an IP address of 192.168.1.100. Specify multiple peers by repeating this command.

- d. Specify which transform sets are allowed for this crypto map entry. List multiple transform sets in order of priority (highest priority first). You can specify up to six transform sets.

```
crypto map map-name seq-num set transform-set transform-set-name1
[transform-set-name2, ...transform-set-name6]
```

For example:

```
crypto map mymap 10 set transform-set myset1 myset2
```

In this example, when traffic matches access list 101, the security association can use either “myset1” (first priority) or “myset2” (second priority) depending on which transform set matches the peer’s transform set.

**Step 4** If the specified transform set includes the AH protocol (authentication via MD5-HMAC or SHA-HMAC), set the AH Security Parameter Index (SPI) and key to apply to inbound protected traffic. If the specified transform set includes only the ESP protocol, skip to [Step 6](#).

```
crypto map map-name seq-num set session-key inbound ah spi hex-key-data
```

For example:

```
crypto map mymaptwo 30 set session-key inbound ah 300
123456789A123456789A123456789A123456789A
```

In this example, the IPSec session key for AH protocol is specified within crypto map “mymaptwo” to be used with the inbound protected traffic.

**Step 5** Set the AH SPIs and keys to apply to outbound protected traffic:

```
crypto map map-name seq-num set session-key outbound ah spi hex-key-data
```

For example:

```
crypto map mymaptwo 30 set session-key outbound ah 400
123456789A123456789A123456789A123456789A
```

**Step 6** If the specified transform set includes the ESP protocol, set the ESP SPIs and keys to apply to inbound protected traffic. If the transform set includes an ESP cipher algorithm, specify the cipher keys. If the transform set includes an ESP authenticator algorithm, specify the authenticator keys.

```
crypto map map-name seq-num set session-key inbound esp spi cipher hex-key-data
[authenticator hex-key-data]
```

For example:

```
crypto map mymaptwo 30 set session-key inbound esp 300 cipher 1234567890123456
authenticator 0000111122223333444455556666777788889999
```

**Step 7** Set the ESP SPIs and keys to apply to outbound protected traffic. If the transform set includes an ESP cipher algorithm, specify the cipher keys. If the transform set includes an ESP authenticator algorithm, specify the authenticator keys.

```
crypto map map-name seq-num set session-key outbound esp spi cipher hex-key-data
[authenticator hex-key-data]
```

For example:

```
crypto map mymaptwo 30 set session-key outbound esp 300 cipher abcdefghijklmnop
authenticator 9999888877776666555544443333222211110000
```

**Step 8** Apply a crypto map set to an interface on which the IPSec traffic will be evaluated:

```
crypto map map-name interface interface-name
```

For example:

```
crypto map mymap interface outside
```

In this example, the PIX Firewall will evaluate the traffic going through the outside interface against the crypto map “mymap” to determine whether it needs to be protected.

**Step 9** Specify that IPSec traffic be implicitly trusted (permitted):

```
sysopt connection permit-ipsec
```



**Note**

This command also permits L2TP/IPSec traffic.

## Viewing IPsec Configuration

Table 6-2 lists commands you can use to view information about your IPsec configuration.

**Table 6-2** Commands to View IPsec Configuration Information

Command	Purpose
<code>show crypto ipsec transform-set</code>	View your transform set configuration.
<code>show crypto map [interface <i>interface-name</i>   tag <i>map-name</i>]</code>	View your crypto map configuration.
<code>show crypto ipsec sa [map <i>map-name</i>   address   identity] [detail]</code>	View information about IPsec security associations.
<code>show crypto dynamic-map [tag <i>map-name</i>]</code>	View information about dynamic crypto maps.
<code>show crypto ipsec security-association lifetime</code>	View global security association lifetime values.

## Clearing SAs

Certain configuration changes will only take effect when negotiating subsequent security associations. If you want the new settings to take immediate effect, clear the existing security associations so that they will be re-established with the changed configuration. For manually established security associations, clear and reinitialize the security associations or the changes will never take effect. If the PIX Firewall is actively processing IPsec traffic, it is desirable to clear only the portion of the security association database that would be affected by the configuration changes (that is, clear only the security associations established by a given crypto map set). Clearing the full security association database should be reserved for large-scale changes, or when the PIX Firewall is processing a small number of other IPsec traffic.

Table 6-3 lists commands you can use to clear and reinitialize IPsec security associations.

**Table 6-3** Commands to Clear and Reinitialize IPsec SAs

Command	Purpose
<code>crypto map <i>map-name</i> interface <i>interface-name</i></code>	Reinitialize the IPsec run-time security association database and security policy database.
<code>clear [crypto] ipsec sa</code> or <code>clear [crypto] ipsec sa peer <i>ip-address</i>   <i>peer-name</i></code> or <code>clear [crypto] ipsec sa map <i>map-name</i></code> or <code>clear [crypto] ipsec sa entry <i>destination-address</i> <i>protocol spi</i></code>	Clear IPsec security associations.  <b>Note</b> Using the <code>clear [crypto] ipsec sa</code> command without parameters will clear out the full security association database, which will clear out active security sessions. You may also specify the <b>peer</b> , <b>map</b> , or <b>entry</b> keywords to clear out only a subset of the security association database. For more information, see the <code>clear [crypto] ipsec sa</code> command within the <i>Cisco PIX Firewall Command Reference</i> .



