



CHAPTER 73

Configuring LAN-to-LAN IPsec VPNs

A LAN-to-LAN VPN connects networks in different geographic locations.

The ASA supports LAN-to-LAN VPN connections to Cisco or third-party peers when the two peers have IPv4 inside and outside networks (IPv4 addresses on the inside and outside interfaces).

For LAN-to-LAN connections using mixed IPv4 and IPv6 addressing, or all IPv6 addressing, the security appliance supports VPN tunnels if both peers are Cisco ASA 5500 series security appliances, and if both inside networks have matching addressing schemes (both IPv4 or both IPv6).

Specifically, the following topologies are supported when both peers are Cisco ASA 5500 series ASAs:

- The ASAs have IPv4 inside networks and the outside network is IPv6 (IPv4 addresses on the inside interfaces and IPv6 addresses on the outside interfaces).
- The ASAs have IPv6 inside networks and the outside network is IPv4 (IPv6 addresses on the inside interface and IPv4 addresses on the outside interfaces).
- The ASAs have IPv6 inside networks and the outside network is IPv6 (IPv6 addresses on the inside and outside interfaces).



Note

The ASA supports LAN-to-LAN IPsec connections with Cisco peers, and with third-party peers that comply with all relevant standards.

This chapter describes how to build a LAN-to-LAN VPN connection. It includes the following sections:

- [Summary of the Configuration, page 73-1](#)
- [Configuring Interfaces, page 73-2](#)
- [Configuring ISAKMP Policy and Enabling ISAKMP on the Outside Interface, page 73-3](#)
- [Creating an IKEv1 Transform Set, page 73-5](#)
- [Creating an IKEv2 Proposal, page 73-6](#)
- [Configuring an ACL, page 73-7](#)
- [Defining a Tunnel Group, page 73-7](#)
- [Creating a Crypto Map and Applying It To an Interface, page 73-9](#)

Summary of the Configuration

This section provides a summary of the example LAN-to-LAN configuration this chapter describes. Later sections provide step-by-step instructions.

```

hostname(config)# interface ethernet0/0
hostname(config-if)# ip address 10.10.4.100 255.255.0.0
hostname(config-if)# nameif outside
hostname(config-if)# no shutdown
hostname(config)# crypto ikev1 policy 1
hostname(config-ikev1-policy)# authentication pre-share
hostname(config-ikev1-policy)# encryption 3des
hostname(config-ikev1-policy)# hash sha
hostname(config-ikev1-policy)# group 2
hostname(config-ikev1-policy)# lifetime 43200
hostname(config)# crypto ikev1 enable outside
hostname(config)# crypto ikev2 policy 1
hostname(config-ikev2-policy)# encryption 3des
hostname(config-ikev2-policy)# group 2
hostname(config-ikev2-policy)# prf sha
hostname(config-ikev2-policy)# lifetime 43200
hostname(config)# crypto ikev2 enable outside
hostname(config)# crypto ipsec ikev1 transform-set FirstSet esp-3des esp-md5-hmac
hostname(config)# crypto ipsec ikev2 ipsec-proposal secure
hostname(config-ipsec-proposal)# protocol esp encryption 3des aes des
hostname(config-ipsec-proposal)# protocol esp integrity sha-1
hostname(config)# access-list 121_list extended permit ip 192.168.0.0 255.255.0.0
150.150.0.0 255.255.0.0
hostname(config)# tunnel-group 10.10.4.108 type ipsec-l2l
hostname(config)# tunnel-group 10.10.4.108 ipsec-attributes
hostname(config-tunnel-ipsec)# ikev1 pre-shared-key 44kkaol59636jnfxf
hostname(config)# crypto map abcmap 1 match address 121_list
hostname(config)# crypto map abcmap 1 set peer 10.10.4.108
hostname(config)# crypto map abcmap 1 set ikev1 transform-set FirstSet
hostname(config)# crypto map abcmap 1 set ikev2 ipsec-proposal secure
hostname(config)# crypto map abcmap interface outside
hostname(config)# write memory

```

Configuring Interfaces

An ASA has at least two interfaces, referred to here as outside and inside. Typically, the outside interface is connected to the public Internet, while the inside interface is connected to a private network and is protected from public access.

To begin, configure and enable two interfaces on the ASA. Then, assign a name, IP address and subnet mask. Optionally, configure its security level, speed, and duplex operation on the security appliance.

To configure interfaces, perform the following steps, using the command syntax in the examples:

-
- Step 1** To enter Interface configuration mode, in global configuration mode enter the **interface** command with the default name of the interface to configure. In the following example the interface is ethernet0.

```

hostname(config)# interface ethernet0/0
hostname(config-if)#

```

- Step 2** To set the IP address and subnet mask for the interface, enter the **ip address** command. In the following example the IP address is 10.10.4.100 and the subnet mask is 255.255.0.0.

```

hostname(config-if)# ip address 10.10.4.100 255.255.0.0
hostname(config-if)#

```

- Step 3** To name the interface, enter the **nameif** command, maximum of 48 characters. You cannot change this name after you set it. In the following example the name of the ethernet0 interface is outside.

```

hostname(config-if)# nameif outside

```

```
hostname(config-if)##
```

- Step 4** To enable the interface, enter the **no** version of the **shutdown** command. By default, interfaces are disabled.

```
hostname(config-if)# no shutdown  
hostname(config-if)#
```

- Step 5** To save your changes, enter the **write memory** command.

```
hostname(config-if)# write memory  
hostname(config-if)#
```

- Step 6** To configure a second interface, use the same procedure.
-

Configuring ISAKMP Policy and Enabling ISAKMP on the Outside Interface

ISAKMP is the negotiation protocol that lets two hosts agree on how to build an IPsec security association (SA). It provides a common framework for agreeing on the format of SA attributes. This includes negotiating with the peer about the SA, and modifying or deleting the SA. ISAKMP separates negotiation into two phases: Phase 1 and Phase 2. Phase 1 creates the first tunnel, which protects later ISAKMP negotiation messages. Phase 2 creates the tunnel that protects data.

IKE uses ISAKMP to setup the SA for IPsec to use. IKE creates the cryptographic keys used to authenticate peers.

The ASA supports IKEv1 for connections from the legacy Cisco VPN client, and IKEv2 for the AnyConnect VPN client.

To set the terms of the ISAKMP negotiations, you create an IKE policy, which includes the following:

- The authentication type required of the IKEv1 peer, either RSA signature using certificates or preshared key (PSK).
- An encryption method, to protect the data and ensure privacy.
- A Hashed Message Authentication Codes (HMAC) method to ensure the identity of the sender, and to ensure that the message has not been modified in transit.
- A Diffie-Hellman group to determine the strength of the encryption-key-determination algorithm. The ASA uses this algorithm to derive the encryption and hash keys.
- For IKEv2, a separate pseudo-random function (PRF) used as the algorithm to derive keying material and hashing operations required for the IKEv2 tunnel encryption, etc.
- A limit to the time the ASA uses an encryption key before replacing it.

With IKEv1 policies, for each parameter, you set one value. For IKEv2, you can configure multiple encryption and authentication types, and multiple integrity algorithms for a single policy. The ASA orders the settings from the most secure to the least secure and negotiates with the peer using that order. This allows you to potentially send a single proposal to convey all the allowed transforms instead of the need to send each allowed combination as with IKEv1.

The following sections provide procedures for creating IKEv1 and IKEv2 policies and enabling them on an interface:

- [Configuring ISAKMP Policies for IKEv1 Connections, page 73-4](#)

- [Configuring ISAKMP Policies for IKEv2 Connections, page 73-4](#)

Configuring ISAKMP Policies for IKEv1 Connections

To configure ISAKMP policies for IKEv1 connections, use the **crypto ikev1 policy** command to enter IKEv1 policy configuration mode where you can configure the IKEv1 parameters:

crypto ikev1 policy *priority*

Perform the following steps and use the command syntax in the following examples as a guide.

-
- Step 1** Enter IPsec IKEv1 policy configuration mode. For example:
- ```
hostname(config)# crypto ikev1 policy 1
hostname(config-ikev1-policy)#
```
- Step 2** Set the authentication method. The following example configures a preshared key:
- ```
hostname(config-ikev1-policy)# authentication pre-share
hostname(config-ikev1-policy)#
```
- Step 3** Set the encryption method. The following example configures 3DES:
- ```
hostname(config-ikev1-policy)# encryption 3des
hostname(config-ikev1-policy)#
```
- Step 4** Set the HMAC method. The following example configures SHA-1:
- ```
hostname(config-ikev1-policy)# hash sha
hostname(config-ikev1-policy)#
```
- Step 5** Set the Diffie-Hellman group. The following example configures Group 2:
- ```
hostname(config-ikev1-policy)# group 2
hostname(config-ikev1-policy)#
```
- Step 6** Set the encryption key lifetime. The following example configures 43,200 seconds (12 hours):
- ```
hostname(config-ikev1-policy)# lifetime 43200
hostname(config-ikev1-policy)#
```
- Step 7** Enable IKEv1 on the interface named outside:
- ```
hostname(config)# crypto ikev1 enable outside
hostname(config)#
```
- Step 8** To save your changes, enter the **write memory** command:
- ```
hostname(config)# write memory
hostname(config)#
```
-

Configuring ISAKMP Policies for IKEv2 Connections

To configure ISAKMP policies for IKEv2 connections, use the **crypto ikev2 policy** command to enter IKEv2 policy configuration mode where you can configure the IKEv2 parameters:

crypto ikev2 policy *priority*

Perform the following steps and use the command syntax in the following examples as a guide:

-
- Step 1** Enter IPsec IKEv2 policy configuration mode. For example:
- ```
hostname(config)# crypto ikev2 policy 1
hostname(config-ikev2-policy)#
```
- Step 2** Set the encryption method. The following example configures 3DES:
- ```
hostname(config-ikev2-policy)# encryption 3des
hostname(config-ikev2-policy)#
```
- Step 3** Set the Diffie-Hellman group. The following example configures Group 2:
- ```
hostname(config-ikev2-policy)# group 2
hostname(config-ikev2-policy)#
```
- Step 4** Set the pseudo-random function (PRF) used as the algorithm to derive keying material and hashing operations required for the IKEv2 tunnel encryption. The following example configures SHA-1 (an HMAC variant):
- ```
hostname(config-ikev2-policy)# prf sha
hostname(config-ikev2-policy)#
```
- Step 5** Set the encryption key lifetime. The following example configures 43,200 seconds (12 hours):
- ```
hostname(config-ikev2-policy)# lifetime 43200
hostname(config-ikev2-policy)#
```
- Step 6** Enable IKEv2 on the interface named outside:
- ```
hostname(config)# crypto ikev2 enable outside
hostname(config)#
```
- Step 7** To save your changes, enter the **write memory** command:
- ```
hostname(config)# write memory
hostname(config)#
```

## Creating an IKEv1 Transform Set

An IKEv1 transform set combines an encryption method and an authentication method. During the IPsec security association negotiation with ISAKMP, the peers agree to use a particular transform set to protect a particular data flow. The transform set must be the same for both peers.

A transform set protects the data flows for the access list specified in the associated crypto map entry. You can create transform sets in the ASA configuration, and then specify a maximum of 11 of them in a crypto map or dynamic crypto map entry.

[Table 73-1](#) lists valid encryption and authentication methods.

**Table 73-1** Valid Encryption and Authentication Methods

| Valid Encryption Methods     | Valid Authentication Methods |
|------------------------------|------------------------------|
| esp-des                      | esp-md5-hmac                 |
| esp-3des (default)           | esp-sha-hmac (default)       |
| esp-aes (128-bit encryption) |                              |
| esp-aes-192                  |                              |

**Table 73-1** Valid Encryption and Authentication Methods

| Valid Encryption Methods | Valid Authentication Methods |
|--------------------------|------------------------------|
| esp-aes-256              |                              |
| esp-null                 |                              |

Tunnel Mode is the usual way to implement IPsec between two ASAs that are connected over an untrusted network, such as the public Internet. Tunnel mode is the default and requires no configuration.

To configure a transform set, perform the following steps:

- Step 1** In global configuration mode enter the **crypto ipsec ikev1 transform-set** command. The following example configures a transform set with the name FirstSet, esp-3des encryption, and esp-md5-hmac authentication. The syntax is as follows:

**crypto ipsec ikev1 transform-set** *transform-set-name encryption-method authentication-method*

```
hostname(config)# crypto ipsec transform-set FirstSet esp-3des esp-md5-hmac
hostname(config)#
```

- Step 2** Save your changes.

```
hostname(config)# write memory
hostname(config)#
```

## Creating an IKEv2 Proposal

For IKEv2, you can configure multiple encryption and authentication types, and multiple integrity algorithms for a single policy. The ASA orders the settings from the most secure to the least secure and negotiates with the peer using that order. This allows you to potentially send a single proposal to convey all the allowed transforms instead of the need to send each allowed combination as with IKEv1.

[Table 73-1](#) lists valid IKEv2 encryption and authentication methods.

**Table 73-2** Valid IKEv2 Encryption and Integrity Methods

| Valid Encryption Methods | Valid Integrity Methods |
|--------------------------|-------------------------|
| des                      | sha (default)           |
| 3des (default)           | md5                     |
| aes                      |                         |
| aes-192                  |                         |
| aes-256                  |                         |

To configure an IKEv2 proposal, perform the following steps:

- Step 1** In global configuration mode, use the **crypto ipsec ikev2 ipsec-proposal** command to enter ipsec proposal configuration mode where you can specify multiple encryption and integrity types for the proposal. In this example, *secure* is the name of the proposal:

```
hostname(config)# crypto ipsec ikev2 ipsec-proposal secure
```

```
hostname(config-ipsec-proposal)#
```

- Step 2** Then enter a protocol and encryption types. ESP is the only supported protocol. For example:

```
hostname(config-ipsec-proposal)# protocol esp encryption 3des aes des
hostname(config-ipsec-proposal)#
```

- Step 3** Enter an integrity type. For example:

```
hostname(config-ipsec-proposal)# protocol esp integrity sha-1
hostname(config-ipsec-proposal)#
```

- Step 4** Save your changes.

## Configuring an ACL

The adaptive security appliance uses access control lists to control network access. By default, the adaptive security appliance denies all traffic. You need to configure an ACL that permits traffic. For more information, see [Chapter 14, “Information About Access Lists.”](#)

The ACLs that you configure for this LAN-to-LAN VPN control connections are based on the source and translated destination IP addresses. Configure ACLs that mirror each other on both sides of the connection.

An ACL for VPN traffic uses the translated address. For more information, see the [“IP Addresses Used for Access Lists When You Use NAT” section on page 14-3.](#)

To configure an ACL, perform the following steps:

- Step 1** Enter the **access-list extended** command. The following example configures an ACL named `l2l_list` that lets traffic from IP addresses in the 192.168.0.0 network travel to the 150.150.0.0 network. The syntax is **access-list listname extended permit ip source-ipaddress source-netmask destination-ipaddress destination-netmask**.

```
hostname(config)# access-list l2l_list extended permit ip 192.168.0.0 255.255.0.0
150.150.0.0 255.255.0.0
hostname(config)#
```

- Step 2** Configure an ACL for the ASA on the other side of the connection that mirrors the ACL above. In the following example the prompt for the peer is `hostname2`.

```
hostname2(config)# access-list l2l_list extended permit ip 150.150.0.0 255.255.0.0
192.168.0.0 255.255.0.0
hostname2(config)#
```



### Note

For more information on configuring an ACL with a `vpn-filter`, see [“Configuring VPN-Specific Attributes” section on page 67-42.](#)

## Defining a Tunnel Group

A tunnel group is a set of records that contain tunnel connection policies. You configure a tunnel group to identify AAA servers, specify connection parameters, and define a default group policy. The ASA stores tunnel groups internally.

There are two default tunnel groups in the ASA: DefaultRAGroup, which is the default IPsec remote-access tunnel group, and DefaultL2Lgroup, which is the default IPsec LAN-to-LAN tunnel group. You can modify them but not delete them.

You can also create one or more new tunnel groups to suit your environment. The ASA uses these groups to configure default tunnel parameters for remote access and LAN-to-LAN tunnel groups when there is no specific tunnel group identified during tunnel negotiation.

To establish a basic LAN-to-LAN connection, you must set two attributes for a tunnel group:

- Set the connection type to IPsec LAN-to-LAN.
- Configure an authentication method for the IP, in the following example, preshared key for IKEv1 and IKEv2.

**Note**

To use VPNs, including tunnel groups, the ASA must be in single-routed mode. The commands to configure tunnel-group parameters do not appear in any other mode.

**Step 1**

To set the connection type to IPsec LAN-to-LAN, enter the **tunnel-group** command. The syntax is **tunnel-group name type type**, where *name* is the name you assign to the tunnel group, and *type* is the type of tunnel. The tunnel types as you enter them in the CLI are:

- **remote-access** (IPsec, SSL, and clientless SSL remote access)
- **ipsec-l2l** (IPsec LAN to LAN)

In the following example the name of the tunnel group is the IP address of the LAN-to-LAN peer, 10.10.4.108.

```
hostname(config)# tunnel-group 10.10.4.108 type ipsec-l2l
hostname(config)#
```

**Note**

LAN-to-LAN tunnel groups that have names that are not an IP address can be used only if the tunnel authentication method is Digital Certificates and/or the peer is configured to use Aggressive Mode.

**Step 2**

To set the authentication method to preshared key, enter the ipsec-attributes mode and then enter the **pre-shared-key** command to create the preshared key. You need to use the same preshared key on both ASAs for this LAN-to-LAN connection.

The key is an alphanumeric string of 1-128 characters.

In the following example the IKEv1 preshared key is 44kkaol59636jnfx:

```
hostname(config)# tunnel-group 10.10.4.108 ipsec-attributes
hostname(config-tunnel-ipsec)# pre-shared-key 44kkaol59636jnfx
```

In the next example, the IKEv2 preshared key is configured also as 44kkaol59636jnfx:

```
hostname(config-tunnel-ipsec)# ikev2 local-authentication pre-shared-key 44kkaol59636jnfx
```

**Step 3**

Save your changes.

```
hostname(config)# write memory
hostname(config)#
```



# Creating a Crypto Map and Applying It To an Interface

Crypto map entries pull together the various elements of IPsec security associations, including the following:

- Which traffic IPsec should protect, which you define in an access list.
- Where to send IPsec-protected traffic, by identifying the peer.
- What IPsec security applies to this traffic, which a transform set specifies.
- The local address for IPsec traffic, which you identify by applying the crypto map to an interface.

For IPsec to succeed, both peers must have crypto map entries with compatible configurations. For two crypto map entries to be compatible, they must, at a minimum, meet the following criteria:

- The crypto map entries must contain compatible crypto access lists (for example, mirror image access lists). If the responding peer uses dynamic crypto maps, the entries in the ASA crypto access list must be “permitted” by the peer’s crypto access list.
- The crypto map entries each must identify the other peer (unless the responding peer is using a dynamic crypto map).
- The crypto map entries must have at least one transform set in common.

If you create more than one crypto map entry for a given interface, use the sequence number (seq-num) of each entry to rank it: the lower the seq-num, the higher the priority. At the interface that has the crypto map set, the ASA evaluates traffic against the entries of higher priority maps first.

Create multiple crypto map entries for a given interface if either of the following conditions exist:

- Different peers handle different data flows.
- 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, define the different types of traffic in two separate access lists, and create a separate crypto map entry for each crypto access list.

To create a crypto map and apply it to the outside interface in global configuration mode, enter several of the **crypto map** commands. These commands use a variety of arguments, but the syntax for all of them begin with **crypto map map-name-seq-num**. In the following example the map-name is **abcmap**, the sequence number is 1.

Enter these commands in global configuration mode:

---

**Step 1** To assign an access list to a crypto map entry, enter the **crypto map match address** command.

The syntax is **crypto map map-name seq-num match address aclname**. In the following example the map name is **abcmap**, the sequence number is 1, and the access list name is **121\_list**.

```
hostname(config)# crypto map abcmap 1 match address 121_list
hostname(config)#
```

**Step 2** To identify the peer (s) for the IPsec connection, enter the **crypto map set peer** command.

The syntax is **crypto map map-name seq-num set peer {ip\_address1 | hostname1} [... ip\_address10 | hostname10]**. In the following example the peer name is 10.10.4.108.

```
hostname(config)# crypto map abcmap 1 set peer 10.10.4.108
hostname(config)#
```

- Step 3** To specify an IKEv1 transform set for a crypto map entry, enter the **crypto map ikev1 set transform-set** command.

The syntax is **crypto map map-name seq-num ikev1 set transform-set transform-set-name**.  
In the following example the transform set name is *FirstSet*.

```
hostname(config)# crypto map abcmap 1 set transform-set FirstSet
hostname(config)#
```

- Step 4** To specify an IKEv2 proposal for a crypto map entry, enter the **crypto map ikev2 set ipsec-proposal** command:

The syntax is **crypto map map-name seq-num set ikev2 ipsec-proposal proposal-name**.  
In the following example the proposal name is *secure*.

```
hostname(config)# crypto map abcmap 1 set ikev2 ipsec-proposal secure
hostname(config)#
```

---

## Applying Crypto Maps to Interfaces

You must apply a crypto map set to each interface through which IPsec traffic travels. The ASA supports IPsec on all interfaces. Applying the crypto map set to an interface instructs the ASA to evaluate all interface traffic against the crypto map set and to use the specified policy during connection or security association negotiations.

Binding a crypto map to an interface also initializes the runtime data structures, such as the security association database and the security policy database. When you later modify a crypto map in any way, the ASA automatically applies the changes to the running configuration. It drops any existing connections and reestablishes them after applying the new crypto map.

- Step 1** To apply the configured crypto map to the outside interface, enter the **crypto map interface** command. The syntax is **crypto map map-name interface interface-name**.

```
hostname(config)# crypto map abcmap interface outside
hostname(config)#
```

- Step 2** Save your changes.

```
hostname(config)# write memory
hostname(config)#
```

---