CHAPTER 29

Configuring LAN-to-LAN IPsec VPNs

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

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

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

```
hostname(config)# interface ethernet0
hostname(config-if)# ip address 10.10.4.100 255.255.0.0
hostname(config-if)# no shutdown
hostname(config)# isakmp policy 1 authentication pre-share
hostname(config)# isakmp policy 1 encryption 3des
hostname(config)# isakmp policy 1 hash sha
hostname(config)# isakmp policy 1 group 2
hostname(config)# isakmp policy 1 lifetime 43200
hostname(config)# isakmp enable outside
hostname(config)# crypto ipsec transform set FirstSet esp-3des esp-md5-hmac
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-ipsec)# pre-shared-key 44kkaol59636jnfx
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 transform-set FirstSet
```

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

Configuring Interfaces

A security appliance 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 security appliance. 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
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

The Internet Security Association and Key Management Protocol, also called IKE, is the negotiation protocol that lets two hosts agree on how to build an IPsec security association. Each ISAKMP negotiation is divided into two sections called Phase1 and Phase 2.

Phase 1 creates the first tunnel, which protects later ISAKMP negotiation messages. Phase 2 creates the tunnel that protects data travelling across the secure connection.

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

- An authentication method, to ensure the identity of the peers.
- An encryption method, to protect the data and ensure privacy.
- A Hashed Message Authentication Codes 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 establish the strength of the encryption-key-determination algorithm. The security appliance uses this algorithm to derive the encryption and hash keys.
- A time limit for how long the security appliance uses an encryption key before replacing it.

See on page 24-3 in the "Configuring IPsec and ISAKMP" chapter of this guide for detailed information about the IKE policy keywords and their values.

To configure ISAKMP policies, in global configuration mode use the **isakmp policy** command with its various arguments. The syntax for ISAKMP policy commands is as follows:

isakmp policy *priority* **attribute_name** [**attribute_value** | *integer*].

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

Step 1 Set the authentication method. The following example configures a preshared key. The priority is 1 in this and all following steps.

```
hostname(config)# isakmp policy 1 authentication pre-share
hostname(config)#
```

Step 2 Set the encryption method. The following example configures 3DES.

```
hostname(config)# isakmp policy 1 encryption 3des
hostname(config)#
```

Step 3 Set the HMAC method. The following example configures SHA-1.

```
hostname(config)# isakmp policy 1 hash sha
hostname(config)#
```

Step 4 Set the Diffie-Hellman group. The following example configures Group 2.

```
hostname(config)# isakmp policy 1 group 2
hostname(config)#
```

Step 5 Set the encryption key lifetime. The following example configures 43,200 seconds (12 hours).

```
hostname(config)# isakmp policy 1 lifetime 43200
hostname(config)#
```

Step 6 Enable ISAKMP on the interface named outside.

```
hostname(config)# isakmp enable outside
hostname(config)#
```

Step 7 To save your changes, enter the **write memory** command.

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

Creating a Transform Set

A 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.

You can create multiple transform sets, and then specify one or more of these transform sets in a crypto map entry. The security appliance uses the transform set to protect the data flows for that crypto map entry access list.

Table 29-1 lists valid encryption and authentication methods.

Table 29-1 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	
esp-aes-256	
esp-null	

Tunnel Mode is the usual way to implement IPsec between two security appliances 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 transform-set** command.

Step 2 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 transform-set *transform-set-name* ***encryption-method authentication-method***

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

Step 3 Save your changes.

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

Configuring an ACL

The security appliance uses access control lists to control network access. By default, the security appliance denies all traffic. You need to configure an ACL that permits traffic. For more information about ACLs, see Chapter 13, “Identifying Traffic with Access Lists.”

Note The ACLs that you configure for this LAN-to-LAN VPN control connections are based on the source IP address and translated destination IP address. For more information about creating an ACL for VPN, see the "IP Addresses Used for Access Lists When You Use NAT" section on page 13-3.

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 security appliance 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
hostname(config)#
```

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 security appliance stores tunnel groups internally.

There are two default tunnel groups in the security appliance system: 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 security appliance 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, in the following example, preshared key.

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:

- **ipsec-ra** (IPsec 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)#
```

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 security appliances for this LAN-to-LAN connection.

The key is an alphanumeric string of 1-128 characters. In the following example the preshared key is 44kkaol59636jnfx.

```
hostname(config)# tunnel-group 10.10.4.108 ipsec-attributes
hostname(config-ipsec)# 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 security appliance 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 security appliance 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 a transform set for a crypto map entry, enter the **crypto map set transform-set** command.

The syntax is **crypto map** *map-name seq-num* **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)#
```

Applying Crypto Maps to Interfaces

You must apply a crypto map set to each interface through which IPsec traffic travels. The security appliance supports IPsec on all interfaces. Applying the crypto map set to an interface instructs the security appliance 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 security appliance 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)#
```