



Configuring Firewall Services

This chapter describes how to configure firewall services and contains these sections:

- [Configuring Firewall Failover, page 4-1](#)
- [Using SNMP, page 4-7](#)
- [Configuring OSPF Routing Support, page 4-15](#)
- [Configuring IPSec for Management, page 4-28](#)

Configuring Firewall Failover

Failover uses two modules that must have identical configurations. You can configure the modules in the following ways:

- An intra-switch failover where two or more firewall modules are in a single chassis.
- An inter-switch failover with a firewall module in each of two chassis.



Note

Refer to the [“Configuring Failover”](#) section on [page 4-4](#) section for a detailed firewall failover configuration description.

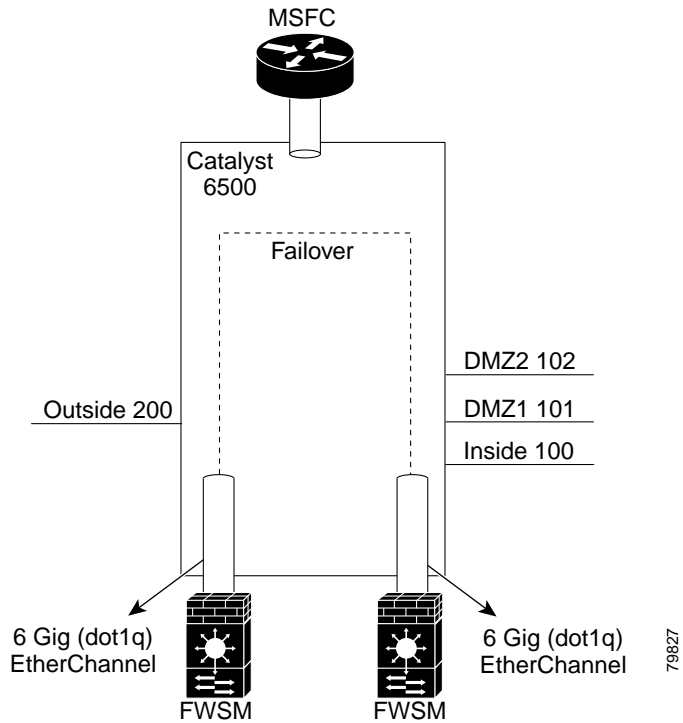
This section describes how to configure failover on the Firewall Services Module:

- [Setting up a Single-Chassis Configuration, page 4-1](#)
- [Setting Up a Dual-Chassis Configuration, page 4-3](#)
- [Configuring Failover, page 4-4](#)

Setting up a Single-Chassis Configuration

To set up failover on a single chassis, install two firewall modules on the same chassis and assign the same firewall VLAN group to both modules.

Figure 4-1 Failover Single Chassis Configuration



To configure failover in a single chassis, perform this task:

Command	Purpose
Step 1 Router(config)# firewall vlan-group <i>group-name</i> <i>vlan-group</i>	Assigns VLANs to a VLAN group.
Step 2 Router(config)# firewall module slot <i>vlan-group</i> <i>group-name</i>	Assigns the VLAN group to the primary module.
Step 3 Router(config)# failover lan interface <i>if_name</i>	Configures the failover interface on the secondary module.
Step 4 Router(config)# firewall module slot <i>vlan-group</i> <i>group-name</i>	Assigns the VLAN group to the secondary module.

This example shows how to configure failover in a single chassis:

```
Router(config)# firewall vlan-group 10 10,20,30,40,50
Router(config)# firewall module 4 vlan-group 10
Router(config)# failover lan interface 1
Router(config)# firewall module 6 vlan-group 10
```

Setting Up a Dual-Chassis Configuration

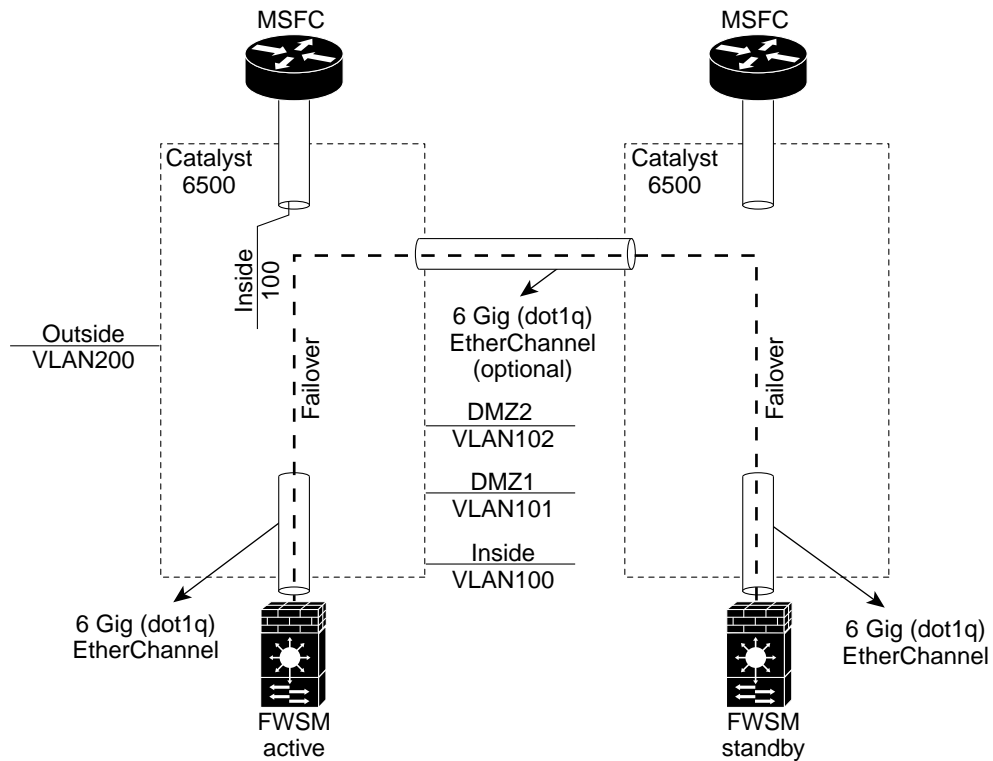
To set up failover across two chassis, install a firewall module in each chassis and assign the same firewall VLAN group to both modules.

To set up a dual-chassis configuration, perform this task:

	Command	Purpose
Step 1	Router1(config)# firewall vlan-group <i>group-name</i> <i>vlan-group</i>	Configures the same set of firewall VLANs on both chassis.
Step 2	Router2(config)# firewall module slot <i>vlan-group</i> <i>group-name</i>	Provides a trunk connecting the two chassis, carrying all the firewall VLANs.

Figure 4-2 shows a dual-chassis configuration.

Figure 4-2 Dual-Chassis Failover Configuration



This example shows how to configure failover in two chassis:

```
Router1(config)# firewall vlan-group 10 10,20,30,40
Router1(config)# firewall module 4 vlan-group 10
Router2(config)# firewall vlan-group 20 10,20,30,40
Router2(config)# firewall module 5 vlan-group 20
```

77118

Configuring Failover

For a failover configuration, both firewall modules need to have the same RAM and Flash memory size and be running the same software version.

To configure failover, follow these steps:

Step 1 Set up one module as the primary with a firewall configuration without failover.



Note Do not add a firewall configuration on the secondary module because a configuration set on the secondary module is not synchronized to the active module. This configuration is cleared during the configuration synchronization from the active module.

Step 2 Create a dedicated logical interface (VLAN) for failover communication using the **nameif** *vlan_id if_name security_level* command.



Note You must add the dedicated logical VLAN to the VLAN group using the **firewall vlan-group** command and activate the dedicated VLAN using the **VLAN [X] state active** command.

Step 3 Configure the module as primary using the **failover lan unit primary** command.

Step 4 Define the failover interface using the **failover lan interface** *if_name* command.

Step 5 Specify the IP address for the primary failover interface using the **ip address** *if_name ip_addr [mask]* command.

This is the IP address used by the primary module on failover interface

Step 6 Assign the IP addresses for all of the interfaces using the **ip address** *if_name ip_address [mask]* command.

Step 7 Specify the failover IP address for the secondary failover interface using the **failover ip address** *if_name ip_addr* command.

This is the IP address used by the secondary module on failover interface.

Step 8 Assign the failover IP addresses for all of the interfaces using the **failover ip address** *if_name ip_addr* command.

This command specifies the IP address used by the standby module on other firewall interfaces. The client hosts are not expected to use this IP address to communicate to the module.

Step 9 Enable failover on the primary module using the **failover** command.

Step 10 Store the failover configuration on the primary module in the Flash using the **write memory** command.



Note This command is required to ensure that the module comes back online with the failover configuration after a reload (or after a failure recovery).

Step 11 When the primary module becomes the active module (use the **show failover** command to see the status), start the failover configuration on the secondary module.

Step 12 The secondary module should not have a firewall configuration. If you need to clear the configuration on the secondary module, use the **clear configure all** command.

- Step 13** Enter the same set of failover commands on the secondary module, repeating [Step 2](#) through [Step 7](#). However, in [Step 3](#) use the **failover lan unit secondary** command for the secondary module.
- The primary and the secondary module should have the identical failover configuration, except for the failover LAN module configuration as primary and secondary.



Note We recommend that you separate the failover and logical update interfaces into separate links. Packets on the failover link are tagged with a higher priority for QOS. Because stateful traffic can be high in volume, the advantages of prioritizing failover traffic are lost by keeping both the failover link and failover LAN interfaces the same.



Note Make sure both primary and secondary modules have the identical definition for the failover interface.

- Step 14** Use the **ping** command to check the connectivity between the primary and secondary module on the failover interface.
- Enter the **icmp permit 0 0 if_name** command to configure the failover interface to allow the ping to go through the firewall.

- Step 15** Save the failover configuration on Flash using the **write memory** command.

The secondary module should detect the primary module and then switch to standby. The firewall configuration is synchronized from the active module to the standby module.



Warning Configuration replication is not performed from the standby module to the active module. Configurations are no longer synchronized.

- Step 16** Enable failover on the secondary module using the **failover** command.

- Step 17** To enable stateful failover, configure a dedicated interface for stateful failover using the **failover link if_name** command, which allows the state information to synchronize.



Note We recommend that you separate the failover and logical update interfaces into separate links. Packets on the failover link are tagged with a higher priority for QOS. Because stateful traffic can be high in volume, the advantages of prioritizing failover traffic are lost by keeping both the failover link and failover LAN interfaces the same.

These examples show how to configure failover on a pair of Firewall Services Modules.

The modules are located in two different switches. A dedicated VLAN (vlan 4000) is created for the failover protocol. The following conditions apply:

- Most of the configuration is performed on the primary module.
- The primary module is designated using the **failover lan unit primary** command.
- Shortly after the **failover** command is entered, the primary module becomes active.
- On the secondary module, only one interface is named using the **nameif** command. Use the interface that is dedicated to the failover protocol.
- The same IP address is assigned to the dedicated failover interface that you assigned to the primary unit (in this example: 10.40.40.1).

- The same address is assigned that you used on the primary unit with the **failover ip address** command. (in this example: 10.40.40.2).

This example shows how to configure the primary module:

```
FWSM(config)# show vlan
30, 40, 4000
FWSM(config)#
FWSM(config)# fail lan unit pri
FWSM(config)# nameif 4000 fover 50
FWSM(config)# nameif 30 outside 0
FWSM(config)# nameif 40 inside 100
FWSM(config)# ip address fover 10.40.40.1 255.255.255.0
FWSM(config)# ip address inside 10.2.1.1 255.255.255.0
FWSM(config)# ip address outside 10.11.1.2 255.255.255.0
FWSM(config)# fail ip address fover 10.40.40.2 255.255.255.0
FWSM(config)# fail ip address inside 10.2.1.2 255.255.255.0
FWSM(config)# fail ip address outside 10.11.1.3 255.255.255.0
FWSM(config)# fail lan int fover
FWSM(config)# logg on
FWSM(config)# logg monitor 7
FWSM(config)# logg con 7
111008: User 'enable_15' executed the 'logging con 7' command.
FWSM(config)# no logg mess 111008
FWSM(config)# no logg mess 111009
FWSM(config)# fail
105002: (Primary) Enabling failover.
FWSM(config)#
        No Response from Mate. Switching to Active
```

You can begin to configure the standby module at this time:

```
Sync Process Start
Sync Process End
709004: (Primary) End Configuration Replication (ACT)
105003: (Primary) Monitoring on interface 2 waiting
105003: (Primary) Monitoring on interface 1 waiting
105004: (Primary) Monitoring on interface 2 normal
105004: (Primary) Monitoring on interface 1 normal
302010: 0 in use, 0 most used
302010: 0 in use, 0 most used
```

This example shows how to configure the standby or secondary module:

```
FWSM(config)# fail lan unit sec
FWSM(config)# nameif 4000 fover 50
FWSM(config)# ip address fover 10.40.40.1 255.255.255.0
FWSM(config)# fail ip address fover 10.40.40.2 255.255.255.0
FWSM(config)# fail lan int fover
FWSM(config)# fail
FWSM(config)# logg on
FWSM(config)# logg mon 7
FWSM(config)# logg con 7
FWSM(config)# 111008: User 'enable_15' executed the 'logging con 7' command.

        Detected an Active mate. Switching to Standby

        Switching to Standby.

FWSM(config)#
Beginning configuration replication from mate.
This unit is in syncing state. 'failover' command will not be effective at this time
End configuration replication from mate.
709006: (Secondary) End Configuration Replication (STB)
Access Rules Download Complete: Memory Utilization < 1%
```

```

105003: (Secondary) Monitoring on interface 2 waiting
105003: (Secondary) Monitoring on interface 1 waiting
105004: (Secondary) Monitoring on interface 2 normal
105004: (Secondary) Monitoring on interface 1 normal

```

This example shows how to monitor the failover status on the primary and secondary modules:

Primary module:

```

FWSM(config)# show fail
Failover On
Failover unit Primary
Failover LAN Interface fover
Reconnect timeout 0:00:00
Poll frequency 15 seconds
  This host: Primary - Active
    Active time: 29925 (sec)
    Interface outside (10.11.1.2): Normal
    Interface inside (10.2.1.1): Normal
  Other host: Secondary - Standby
    Active time: 285 (sec)
    Interface outside (10.11.1.3): Normal
    Interface inside (10.2.1.2): Normal

Stateful Failover Logical Update Statistics
Link : Unconfigured.

```

Secondary module:

```

FWSM(config)# show fail
Failover On
Failover unit Secondary
Failover LAN Interface fover
Reconnect timeout 0:00:00
Poll frequency 15 seconds
  This host: Secondary - Standby
    Active time: 285 (sec)
    Interface inside (10.2.1.2): Normal
    Interface outside (10.11.1.3): Normal
  Other host: Primary - Active
    Active time: 30750 (sec)
    Interface inside (10.2.1.1): Normal
    Interface outside (10.11.1.2): Normal

Stateful Failover Logical Update Statistics
Link : Unconfigured.

FWSM(config)#

```

Using SNMP

You can monitor system events on the Firewall Services Module by using SNMP. You can read SNMP events, but information on the module cannot be changed with SNMP.

Use CiscoWorks for Windows or any other SNMP V1, MIB-II-compliant browser to receive SNMP traps and browse a MIB. SNMP traps occur at UDP port 162.



Note

The Firewall Services Module does not support browsing of the Cisco syslog MIB.

You can browse the System and Interface groups of MIB-II. Browsing an MIB is different from sending traps. Browsing involves doing an `snmpget` or `snmpwalk` of the MIB tree from the management station to determine values.

This section describes how to use SNMP on the Firewall Services Module:

- [MIB Support, page 4-8](#)
- [SNMP Traps, page 4-8](#)
- [Compiling Cisco Syslog MIB Files, page 4-9](#)
- [Using the Firewall and Memory Pool MIBs, page 4-10](#)
- [SNMP Usage Notes, page 4-15](#)

MIB Support

The Firewall Services Module supports the Cisco Firewall MIB and Cisco Memory Pool MIB.

The Firewall Services Module does not support the following in the Cisco Firewall MIB:

- `cfwSecurityNotification NOTIFICATION-TYPE`
- `cfwContentInspectNotification NOTIFICATION-TYPE`
- `cfwConnNotification NOTIFICATION-TYPE`
- `cfwAccessNotification NOTIFICATION-TYPE`
- `cfwAuthNotification NOTIFICATION-TYPE`
- `cfwGenericNotification NOTIFICATION-TYPE`

SNMP Traps

Traps are unsolicited “comments” from the managed device to the management station for specific events, such as link up, link down, and syslog event generation.

The **`snmp-server`** command causes the Firewall Services Module to send SNMP traps so that the module can be monitored remotely. Use the **`snmp-server host`** command to specify which systems receive the SNMP traps.

An SNMP object ID (OID) for the module displays in SNMP event traps sent from the module. The Firewall Services Module provides the system OID in SNMP event traps and SNMP `mib-2.system.sysObjectID` equal to the (1.3.6.1.4.1.9.1.227) original PIX firewall OID.

The module responds to an SNMP request from a management station and then the module sends an event notification trap.

The Firewall Services Module SNMP traps available to an SNMP management station are as follows:

- Generic traps:
 - Link up and link down (VLAN connected to the interface or not)
 - Cold start
 - Authentication failure (mismatched community string)
- Security-related events are sent through the Cisco Syslog MIB:
 - Global access denied

- Failover syslog messages
- syslog messages

Receiving Requests and Sending Syslog Traps

To receive requests and send traps from the Firewall Services Module to an SNMP management station, follow these steps:

-
- Step 1** Identify the IP address of the SNMP management station by using the **snmp-server host** command.
- Step 2** Set the **snmp-server** options for **location**, **contact**, and the **community** password as required.
- You do not need to do further configuration if you only want to send the cold start, link up, and link down generic traps, and you only want to receive SNMP requests.
- Step 3** Add an **snmp-server enable traps** command statement to the configuration.
- Step 4** Set the logging level with the **logging history** command:
- ```
logging history debugging
```
- We recommend that you use the debugging level during initial setup and during testing. After setup, set the level from debugging to a lower value.
- The **logging history** command sets the severity level for SNMP syslog messages.
- Step 5** Start sending syslog traps to the management station using the **logging on** command.
- Step 6** To disable sending syslog traps, use the **no logging on** command or the **no snmp-server enable traps** command.
- 

## Compiling Cisco Syslog MIB Files

To receive security and failover SNMP traps from the Firewall Services Module, compile the Cisco SMI MIB and the Cisco syslog MIB into your SNMP management application. If you do not compile the Cisco syslog MIB into your application, you receive only traps for link up or down, firewall cold start, and authentication failure.

To obtain the Cisco MIB files go to the following URLs:

- <http://www.cisco.com/public/mibs/v2/CISCO-FIREWALL-MIB.my>
- <ftp://ftp.cisco.com/pub/mibs/v2/CISCO-FIREWALL-MIB.my>
- <http://www.cisco.com/public/mibs/v2/CISCO-MEMORY-POOL-MIB.my>
- <ftp://ftp.cisco.com/pub/mibs/v2/CISCO-MEMORY-POOL-MIB.my>
- <http://www.cisco.com/public/mibs/v2/CISCO-SMI.my>
- <ftp://ftp.cisco.com/pub/mibs/v2/CISCO-SMI.my>
- <http://www.cisco.com/public/mibs/v2/CISCO-SYSLOG-MIB.my>
- <ftp://ftp.cisco.com/pub/mibs/v2/CISCO-SYSLOG-MIB.my>

To compile Cisco syslog MIB files into your browser using CiscoWorks for Windows (SNMPc), follow these steps:

- 
- Step 1 Obtain the Cisco syslog MIB files.
  - Step 2 Start SNMPc.
  - Step 3 Select **Config>Compile MIB**.
  - Step 4 Scroll to the bottom of the list, and select the last entry.
  - Step 5 Click **Add**.
  - Step 6 Find the Cisco syslog MIB files.



**Note** With certain applications, only files with a .mib extension may show in the file selection window of the SNMPc. The Cisco syslog MIB files with the .my extension shown. In this case, you should manually change the .my extension to a .mib extension.

---

- Step 7 Select CISCO-FIREWALL-MIB.my (CISCO-FIREWALL-MIB.mib) and click **OK**.
  - Step 8 Scroll to the bottom of the list, and select the last entry.
  - Step 9 Click **Add**.
  - Step 10 Locate the CISCO-MEMORY-POOL-MIB.my (CISCO-MEMORY-POOL-MIB.mib) file and click **OK**.
  - Step 11 Scroll to the bottom of the list, and click the last entry.
  - Step 12 Click **Add**.
  - Step 13 Locate the CISCO-SMI.my (CISCO-SMI.mib) file and click **OK**.
  - Step 14 Scroll to the bottom of the list, and select the last entry.
  - Step 15 Click **Add**.
  - Step 16 Locate the CISCO-SYSLOG-MIB.my (CISCO-SYSLOG-MIB.mib) file and click **OK**.
  - Step 17 Click **Load All**.
  - Step 18 Restart SNMPc if there are no errors. If there are errors, check your configuration.
- 

## Using the Firewall and Memory Pool MIBs

You can poll failover and system status using the Cisco Firewall and Memory Pool MIBs. With the MIB tables, you can view failover status, memory usage, connection count, and system buffer usage.

### Viewing Failover Status

The Cisco Firewall MIB `cfsHardwareStatusTable` indicates whether failover is enabled, and which module is active. The Cisco Firewall MIB indicates failover status in two rows in the `cfwHardwareStatusTable` object. From the Firewall Services Module command line, you can view failover status using the **show failover** command. You can access the object table from the following path:

```
.iso.org.dod.internet.private.enterprises.cisco.ciscoMgmt.ciscoFirewallMIB.
ciscoFirewallMIBObjects.cfwSystem.cfwStatus.cfwHardwareStatusTable
```

[Table 4-1](#) lists which objects provide failover information.

Table 4-1 Failover Status Objects

| Object                        | Object Type     | Row 1: Returned if Failover is Disabled | Row 1: Returned if Failover is Enabled                                                    | Row 2: Returned if Failover is Enabled                                                    |
|-------------------------------|-----------------|-----------------------------------------|-------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| cfwHardwareType (table index) | Hardware        | <b>6</b> (primary module) <sup>1</sup>  | <b>6</b> (primary module)                                                                 | <b>7</b> (secondary module)                                                               |
| cfwHardwareInformation        | SnmpAdminString | blank                                   | blank                                                                                     | blank                                                                                     |
| cfwHardwareStatusValue        | HardwareStatus  | <b>0</b> (not used)                     | <b>active</b> or <b>9</b> (active module) or <b>standby</b> or <b>10</b> (standby module) | <b>active</b> or <b>9</b> (active module) or <b>standby</b> or <b>10</b> (standby module) |
| cfwHardwareStatusDetail       | SnmpAdminString | <b>Failover Off</b>                     | blank                                                                                     | blank                                                                                     |

1. The type of returned values are shown in parentheses.

In the HP OpenView Browse MIB application's MIB values window, if failover is disabled, a sample MIB query displays the following information:

```
cfwHardwareInformation.6 :
cfwHardwareInformation.7 :
cfwHardwareStatusValue.6 :0
cfwHardwareStatusValue.7 :0
cfwHardwareStatusDetail.6 :Failover Off
cfwHardwareStatusDetail.7 :Failover Off
```

In this list, the table index, cfwHardwareType, appears as either .6 or .7 appended to the end of each of the subsequent objects. The cfwHardwareInformation field is blank, the cfwHardwareStatusValue is 0, and the cfwHardwareStatusDetail contains Failover Off, which indicates the failover status.

When failover is enabled, a sample MIB query displays the following information:

```
cfwHardwareInformation.6 :
cfwHardwareInformation.7 :
cfwHardwareStatusValue.6 : active
cfwHardwareStatusValue.7 : standby
cfwHardwareStatusDetail.6 :
cfwHardwareStatusDetail.7 :
```

In this list, only the cfwHardwareStatusValue contains either active or standby values to indicate the status of each module.

## Verifying Memory Usage

You can determine how much free memory is available with the Cisco Memory Pool MIB. From the Firewall Services Module command line, use the **show memory** command to view the memory usage. The following is sample output from the **show memory** command:

```
Router(config)# show memory
16777216 bytes total, 5595136 bytes free
```

You can access the MIB objects from the following path:

```
.iso.org.dod.internet.private.enterprises.cisco.ciscoMgmt.ciscoMemoryPoolMIB.
ciscoMemoryPoolObjects.ciscoMemoryPoolTable
```

Table 4-2 lists which objects provide memory usage information.

Table 4-2 Memory Usage Objects

| Object                               | Object Type          | Returned Value                                                                  |
|--------------------------------------|----------------------|---------------------------------------------------------------------------------|
| ciscoMemoryPoolType<br>(table index) | CiscoMemoryPoolTypes | 1 (processor memory)                                                            |
| ciscoMemoryPoolName                  | DisplayString        | Firewall Services Module system memory                                          |
| ciscoMemoryPoolAlternate             | Integer32            | 0 (no alternate memory pool)                                                    |
| ciscoMemoryPoolValid                 | TruthValue           | true (the values of the remaining objects are valid)                            |
| ciscoMemoryPoolUsed                  | Gauge32              | integer (number of bytes currently in use—the total bytes minus the free bytes) |
| ciscoMemoryPoolFree                  | Gauge32              | integer (number of bytes currently free)                                        |
| ciscoMemoryPoolLargestFree           | Gauge32              | 0 (information not available)                                                   |

In the HP OpenView Browse MIB application's MIB values window, a sample MIB query displays the following information:

```
ciscoMemoryPoolName.1 :FWSM system memory
ciscoMemoryPoolAlternate.1 :0
ciscoMemoryPoolValid.1 :true
ciscoMemoryPoolUsed.1 :12312576
ciscoMemoryPoolFree.1 :54796288
ciscoMemoryPoolLargestFree.1 :0
```

In this list, the table index, ciscoMemoryPoolName, appears as the .1 value at the end of each subsequent object value. The ciscoMemoryPoolUsed object lists the number of bytes currently in use (12312576) and the ciscoMemoryPoolFree object lists the number of bytes currently free (54796288). The other objects always list the values described in Table 4-2.

## Viewing the Connection Count

You can view the number of connections in use from the cfwConnectionStatTable in the Cisco Firewall MIB. From the Firewall Services Module command line, enter the **show conn** command to view the connection count. The following is sample output from the **show connection** command:

```
show connection count
15 in use
```

The cfwConnectionStatTable object table can be accessed from the following path:

```
.iso.org.dod.internet.private.enterprises.cisco.ciscoMgmt.ciscoFirewallMIB.
ciscoFirewallMIBObjects.cfwSystem.cfwStatistics.cfwConnectionStatTable
```

Table 4-3 lists which objects provide connection count information.

Table 4-3 Connection Count Objects

| Object                                    | Object Type     | Row 1: Returned Value                                                | Row 2: Returned Value                                                            |
|-------------------------------------------|-----------------|----------------------------------------------------------------------|----------------------------------------------------------------------------------|
| cfwConnectionStatService<br>(Table index) | Service         | 40 (IP protocol)                                                     | 40 (IP protocol)                                                                 |
| cfwConnectionStatType<br>(Table index)    | ConnectionStat  | 6 (current connections in use)                                       | 7 (high)                                                                         |
| cfwConnectionStatDescription              | SnmpAdminString | <b>number of connections currently in use by the entire firewall</b> | <b>highest number of connections in use at any one time since system startup</b> |
| cfwConnectionStatCount                    | Counter32       | 0 (not used)                                                         | 0 (not used)                                                                     |
| cfwConnectionStatValue                    | Gauge32         | integer (in use number)                                              | 0 (not used)                                                                     |

In the HP OpenView Browse MIB application's MIB values window, a sample MIB query displays the following information:

```
cfwConnectionStatDescription.40.6 :number of connections currently in use by the entire
firewall
cfwConnectionStatDescription.40.7 :highest number of connections in use at any one time
since system startup
cfwConnectionStatCount.40.6 :0
cfwConnectionStatCount.40.7 :0
cfwConnectionStatValue.40.6 :15
cfwConnectionStatValue.40.7 :15
```

In this list, the table index, cfwConnectionStatService, appears as the .40 appended to each subsequent object. The table index, cfwConnectionStatType, appears as either .6 to indicate the number of connections in use or as .7 to indicate the most used number of connections. The cfwConnectionStatValue object lists the connection count. The cfwConnectionStatCount object always returns 0 (zero).

## Viewing System Buffer Usage

You can view the system buffer usage from the Cisco Firewall MIB in multiple rows of the cfwBufferStatsTable. The system buffer usage provides an early warning that the Firewall Services Module is reaching its capacity limit. On the command line, enter the **show blocks** command to view this information. The following is sample output from the **show blocks** command to demonstrate how cfwBufferStatsTable is populated:

```
show blocks
SIZE MAX LOW CNT
4 1600 1600 1600
80 100 97 97
256 80 79 79
1550 780 402 404
65536 8 8 8
```

You can view cfwBufferStatsTable at the following path:

```
.iso.org.dod.internet.private.enterprises.cisco.ciscoMgmt.ciscoFirewallMIB.
ciscoFirewallMIBObjects.cfwSystem.cfwStatistics.cfwBufferStatsTable
```

Table 4-4 lists the objects required to view the system block usage.

Table 4-4 System Block Usage Objects

| Object                             | Object Type        | First Row: Returned Value                                                                                  | Next Row: Returned Value                                                                                             | Next Row: Returned Value                                                                                   |
|------------------------------------|--------------------|------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
| cfwBufferStatSize<br>(Table index) | Unsigned32         | <i>integer</i> (SIZE value; for example, 4 for a 4-byte block)                                             | <i>integer</i> (SIZE value; for example, 4 for a 4-byte block)                                                       | <i>integer</i> (SIZE value; for example, 4 for a 4-byte block)                                             |
| cfwBufferStatType<br>(Table index) | ResourceStatistics | <b>3</b> (MAX)                                                                                             | <b>5</b> (LOW)                                                                                                       | <b>8</b> (CNT)                                                                                             |
| cfwBufferStatInformation           | SnmpAdminString    | <b>maximum number of allocated integer byte blocks</b> ( <i>integer</i> is the number of bytes in a block) | <b>fewest integer byte blocks available since system startup</b> ( <i>integer</i> is the number of bytes in a block) | current number of <b>available integer byte blocks</b> ( <i>integer</i> is the number of bytes in a block) |
| cfwBufferStatValue                 | Gauge32            | <i>integer</i> (MAX number)                                                                                | <i>integer</i> (LOW number)                                                                                          | (CNT number)                                                                                               |

**Note**

The three rows repeat for every block size listed in the output of the **show blocks** command.

In the HP OpenView Browse MIB application's MIB values window a sample MIB query displays the following information:

```
cfwBufferStatInformation.4.3 :maximum number of allocated 4 byte blocks
cfwBufferStatInformation.4.5 :fewest 4 byte blocks available since system startup
cfwBufferStatInformation.4.8 :current number of available 4 byte blocks
cfwBufferStatInformation.80.3 :maximum number of allocated 80 byte blocks
cfwBufferStatInformation.80.5 :fewest 80 byte blocks available since system startup
cfwBufferStatInformation.80.8 :current number of available 80 byte blocks
cfwBufferStatInformation.256.3 :maximum number of allocated 256 byte blocks
cfwBufferStatInformation.256.5 :fewest 256 byte blocks available since system startup
cfwBufferStatInformation.256.8 :current number of available 256 byte blocks
cfwBufferStatInformation.1550.3 :maximum number of allocated 1550 byte blocks
cfwBufferStatInformation.1550.5 :fewest 1550 byte blocks available since system startup
cfwBufferStatInformation.1550.8 :current number of available 1550 byte blocks
cfwBufferStatValue.4.3: 1600
cfwBufferStatValue.4.5: 1600
cfwBufferStatValue.4.8: 1600
cfwBufferStatValue.80.3: 400
cfwBufferStatValue.80.5: 396
cfwBufferStatValue.80.8: 400
cfwBufferStatValue.256.3: 1000
cfwBufferStatValue.256.5: 997
cfwBufferStatValue.256.8: 999
cfwBufferStatValue.1550.3: 1444
cfwBufferStatValue.1550.5: 928
cfwBufferStatValue.1550.8: 932
```

In this list, the first table index, `cfwBufferStatSize`, appears as first number appended to the end of each object, such as `.4` or `.256`. The other table index, `cfwBufferStatType`, appears as `.3`, `.5`, or `.8` after the first index. For each block size, the `cfwBufferStatInformation` object identifies the type of value and the `cfwBufferStatValue` object identifies the number of bytes for each value.

## Using the ipAddrTable

When you use the SNMP ipAddrTable entry, all interfaces must have unique addresses. If interfaces have not been assigned IP addresses, their IP addresses are all set to 127.0.0.1 by default. Duplicate IP addresses cause the SNMP management station to loop indefinitely. If this situation occurs, assign each interface a different address. For example, you can set one address to 127.0.0.1, another to 127.0.0.2, and so on.

SNMP uses a sequence of GetNext operations to traverse the MIB tree. Each GetNext request is based on the result of the previous request. If two consecutive interfaces have the same IP 127.0.0.1 (table index), the GetNext function returns 127.0.0.1, which is correct. However, when SNMP generates the next GetNext request using the same result (127.0.0.1), the request is identical to the previous one, which causes the management station to loop infinitely. For example:

```
GetNext(ip.ipAddrTable.ipAddrEntry.ipAdEntAddr.127.0.0.1)
```

With SNMP, the MIB table index must be unique for the agent to identify a row from the MIB table. The table index for ip.ipAddrTable is the module interface IP address, which requires that the IP address is unique. The SNMP agent might become confused and may return information of another interface (row), which has the same IP (index).

## SNMP Usage Notes

The following notes apply:

- The MIB-II ifEntry.ifAdminStatus object returns 1 if the interface is accessible. The object returns 2 if you administratively shut down the interface using the **shutdown** option of the **interface** command.
- The SNMP ifOutUcastPkts object now correctly returns the outbound packet count.
- Syslog messages generated by the SNMP module specify the interface name instead of an interface number.
- The **ifSpeed** option is not supported and will always return a zero.

## Configuring OSPF Routing Support

The Firewall Services Module can run two processes of Open Shortest Path First (OSPF) protocol simultaneously. Each of the OSPF processes runs on a different set of interfaces. RIP cannot be enabled on any of the same interfaces as the interfaces that OSPF is enabled on.

Redistribution between the two OSPF processes is supported. Redistribution between RIP and OSPF is not supported in the current release. Static and connected routes configured on OSPF-enabled interfaces on the Firewall Services Module can also be redistributed into the OSPF process. For further information on how to configure OSPF redistribution on the Firewall Services Module, refer to the “Configuring IP Routing Protocol-Independent Features” section in the *Cisco IOS IP and IP Routing Configuration Guide*.

OSPF is not supported in topologies where the same router or networks are connected to two different interfaces of the Firewall Services Module. OSPF does not handle the address translations configured between interfaces. Care should be taken in the network design to not advertise private addresses into the global networks. Use separate OSPF processes or use filtering mechanisms.

OSPF allows the module to maintain its own routing table. The OSPF protocol provides the following features for the module:

- Support of intra-area, interarea, and external (type I and Type II) routes.
- Support of a virtual link being configured on or through the module.
- OSPF link-state advertisement (LSA) flooding.
- Authentication to OSPF packets (both password and MD5 authentication).
- Support for configuring the module as a designated router or a backup designated router. The module also can be set up as an area border router, however, the ability to configure the module as an autonomous system boundary router is limited to default information only (for example, injecting a default route).
- Support for stub areas and not-so-stubby-area (NSSA).
- Area boundary router type-3 LSA filtering.
- Advertisement of static and global address translations.

This section describes how to use OSPF on the Firewall Services Module:

- [Enabling OSPF, page 4-17](#)
- [Configuring OSPF Interface Parameters, page 4-17](#)
- [Configuring OSPF Area Parameters, page 4-18](#)
- [Configuring OSPF NSSA, page 4-19](#)
- [Configuring Route Summarization Between OSPF Areas, page 4-20](#)
- [Configuring Route Summarization when Redistributing Routes into OSPF, page 4-20](#)
- [Creating Virtual Links, page 4-21](#)
- [Generating a Default Route, page 4-21](#)
- [Changing the OSPF Administrative Distances, page 4-22](#)
- [Configuring Route Calculation Timers, page 4-22](#)
- [Logging Neighbors Going Up or Down, page 4-22](#)
- [Changing the LSA Group Pacing, page 4-23](#)
- [Blocking OSPF LSA Flooding, page 4-24](#)
- [Ignoring MOSPF LSA Packets, page 4-25](#)
- [Displaying OSPF Update Packet Pacing, page 4-26](#)
- [Area Border Router Type 3 LSA Filtering, page 4-26](#)
- [Monitoring and Maintaining OSPF, page 4-27](#)

## Enabling OSPF

As with other routing protocols, to enable OSPF you need to create an OSPF routing process, specify the range of IP addresses to be associated with the routing process, and assign area IDs to be associated with that range of IP addresses. To enable OSPF, perform this task, beginning in global configuration mode:

|        | Command                                                                       | Purpose                                                                             |
|--------|-------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| Step 1 | <code>FWSM(config)# <b>router ospf</b> process-id</code>                      | Enables OSPF routing, which places you in router configuration mode.                |
| Step 2 | <code>FWSM(config-router)# <b>network</b> ip-address mask area area-id</code> | Defines an interface on which OSPF runs and defines the area ID for that interface. |

This example shows how to enable OSPF:

```
FWSM(config)# router ospf 2
FWSM(config-router)# network 2.0.0.0 255.0.0.0 area 0
```

## Configuring OSPF Interface Parameters

Cisco OSPF implementation allows you to alter some interface-specific OSPF parameters as necessary. You are not required to alter any of these parameters, but some interface parameters must be consistent across all routers in an attached network. You configure the parameters by using the **ospf hello-interval**, **ospf dead-interval**, and **ospf authentication-key** interface configuration commands. Be sure that if you do configure any of these parameters, the configurations for all routers on your network have compatible values.

To specify interface parameters for your network, perform this task in interface configuration mode:

|        | Command                                                                      | Purpose                                                                                                                                       |
|--------|------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | <code>FWSM(config)# <b>interface</b> interface_name</code>                   | Specifies the OSPF interface.                                                                                                                 |
| Step 2 | <code>FWSM(config-interface)# <b>ospf cost</b> cost</code>                   | Explicitly specifies the cost of sending a packet on an OSPF interface.                                                                       |
| Step 3 | <code>FWSM(config-interface)# <b>ospf retransmit-interval</b> seconds</code> | Specifies the number of seconds between link-state advertisement (LSA) retransmissions for adjacencies belonging to an OSPF interface.        |
| Step 4 | <code>FWSM(config-interface)# <b>ospf transmit-delay</b> seconds</code>      | Sets the estimated number of seconds required to send a link-state update packet on an OSPF interface.                                        |
| Step 5 | <code>FWSM(config-interface)# <b>ospf priority</b> number-value</code>       | Sets priority to help determine the OSPF designated router for a network.                                                                     |
| Step 6 | <code>FWSM(config-interface)# <b>ospf hello-interval</b> seconds</code>      | Specifies the length of time between the hello packets that the Cisco IOS software sends on an OSPF interface.                                |
| Step 7 | <code>FWSM(config-interface)# <b>ospf dead-interval</b> seconds</code>       | Sets the number of seconds that a device must wait before it declares a neighbor OSPF router down because it has not received a hello packet. |
| Step 8 | <code>FWSM(config-interface)# <b>ospf authentication-key</b> key</code>      | Assigns a password to be used by neighboring OSPF routers on a network segment that is using the OSPF simple password authentication.         |

|         | Command                                                                              | Purpose                                                                                                                                                          |
|---------|--------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 9  | FWSM(config-interface)# <b>ospf message-digest-key</b><br><i>key-id md5 key</i>      | Enables OSPF MD5 authentication. The values for the <i>key-id</i> and <i>key</i> arguments must match values specified for other neighbors on a network segment. |
| Step 10 | FWSM(config-interface)# <b>ospf authentication</b><br><b>[message-digest   null]</b> | Specifies the authentication type for an interface.                                                                                                              |
| Step 11 | FWSM(config-interface)# <b>show ip ospf</b>                                          | Displays the OSPF configuration.                                                                                                                                 |

This example shows how to configure the OSPF interfaces:

```
FWSM(config)# router ospf 2
FWSM(config-router)# network 2.0.0.0 255.0.0.0 area 0
FWSM(config-router)# interface inside
FWSM(config-interface)# ospf cost 20
FWSM(config-interface)# ospf retransmit-interval 15
FWSM(config-interface)# ospf transmit-delay 10
FWSM(config-interface)# ospf priority 20
FWSM(config-interface)# ospf hello-interval 10
FWSM(config-interface)# ospf dead-interval 40
FWSM(config-interface)# ospf authentication-key cisco
FWSM(config-interface)# ospf message-digest-key 1 md5 cisco
FWSM(config-interface)# ospf authentication message-digest
FWSM(config-interface)# exit
FWSM(config)# show ip ospf
```

```
Routing Process "ospf 2" with ID 20.1.89.2 and Domain ID 0.0.0.2
Supports only single TOS(TOS0) routes
Supports opaque LSA
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 5. Checksum Sum 0x 26da6
Number of opaque AS LSA 0. Checksum Sum 0x 0
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
External flood list length 0
 Area BACKBONE(0)
 Number of interfaces in this area is 1
 Area has no authentication
 SPF algorithm executed 2 times
 Area ranges are
 Number of LSA 5. Checksum Sum 0x 209a3
 Number of opaque link LSA 0. Checksum Sum 0x 0
 Number of DCbitless LSA 0
 Number of indication LSA 0
 Number of DoNotAge LSA 0
 Flood list length 0
```

## Configuring OSPF Area Parameters

You can configure several area parameters using Cisco OSPF software. These area parameters (shown in the following task table) include authentication, defining stub areas, and assigning specific costs to the default summary route. Authentication provides password-based protection against unauthorized access to an area.

Stub areas are areas into which information on external routes is not sent. Instead, there is a default external route generated by the area border router, into the stub area for destinations outside the autonomous system. To take advantage of the OSPF stub area support, default routing must be used in

the stub area. To further reduce the number of LSAs sent into a stub area, you can configure the no-summary keyword of the area stub router configuration command on the area border router to prevent it from sending summary link advertisement (LSAs type 3) into the stub area.

To specify an area parameter for your network, perform this task in router configuration mode:

|        | Command                                                                | Purpose                                                                      |
|--------|------------------------------------------------------------------------|------------------------------------------------------------------------------|
| Step 1 | FWSM(config-router)# <b>area area-id authentication</b>                | Enables authentication for an OSPF area.                                     |
| Step 2 | FWSM(config-router)# <b>area area-id authentication message-digest</b> | Enables MD5 authentication for an OSPF area.                                 |
| Step 3 | FWSM(config-router)# <b>area area-id stub [no-summary]</b>             | Defines an area to be a stub area.                                           |
| Step 4 | FWSM(config-router)# <b>area area-id default-cost cost</b>             | Assigns a specific cost to the default summary route used for the stub area. |

This example shows how to configure the OSPF area parameters:

```
FWSM(config)# router ospf 2
FWSM(config-router)# area 0 authentication
FWSM(config-router)# area 0 authentication message-digest
FWSM(config-router)# area 17 stub
FWSM(config-router)# area 17 default-cost 20
```

## Configuring OSPF NSSA

The OSPF implementation of NSSA is similar to OSPF stub area. NSSA does not flood type 5 external LSAs from the core into the area, but it can import autonomous system external routes in a limited way within the area.

NSSA imports type 7 autonomous system external routes within an NSSA area by redistribution. These type 7 LSAs are translated into type 5 LSAs by NSSA area border routers, which are flooded throughout the whole routing domain. Summarization and filtering are supported during the translation.

You can simplify administration if you are an Internet service provider (ISP) or a network administrator that must connect a central site using OSPF to a remote site that is using a different routing protocol using NSSA.

Before the implementation of NSSA, the connection between the corporate site border router and the remote router could not be run as OSPF stub area because routes for the remote site could not be redistributed into stub area, and two routing protocols needed to be maintained. A simple protocol such as RIP was usually run and handled the redistribution. With NSSA, you can extend OSPF to cover the remote connection by defining the area between the corporate router and the remote router as an NSSA.

To specify area parameters as needed to configure OSPF NSSA, perform this task in router configuration mode:

| Command                                                                                           | Purpose               |
|---------------------------------------------------------------------------------------------------|-----------------------|
| FWSM(config-router)# <b>area area-id nssa [no-redistribution] [default-information-originate]</b> | Defines an NSSA area. |

This example shows how to define an NSSA area:

```
FWSM(config-router)# area 17 nssa
```

To control summarization and filtering of type 7 LSAs into type 5 LSAs, perform this task in router configuration mode on the area border router:

| Command                                                                                        | Purpose                                                          |
|------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
| <code>FWSM(config-router)# <b>summary</b> address prefix mask [not advertise] [tag tag]</code> | Controls the summarization and filtering during the translation. |

This example shows how to control summarization and filtering:

```
FWSM(config-router)# summary-address 12.1.0.0 255.255.0.0
```

Before you use this feature, consider these guidelines:

- You can set a type 7 default route that can be used to reach external destinations. When configured, the router generates a type 7 default into the NSSA or the NSSA area boundary router.
- Every router within the same area must agree that the area is NSSA; otherwise, the routers will not be able to communicate.

## Configuring Route Summarization Between OSPF Areas

*Route summarization* is the consolidation of advertised addresses. This feature causes a single summary route to be advertised to other areas by an area boundary router. In OSPF, an area boundary router will advertise networks in one area into another area. If the network numbers in an area are assigned in a way such that they are contiguous, you can configure the area boundary router to advertise a summary route that covers all the individual networks within the area that fall into the specified range.

To specify an address range, perform this task in router configuration mode:

| Command                                                                                                        | Purpose                                                                 |
|----------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|
| <code>FWSM(config-router)# <b>area</b> area-id <b>range</b> ip-address mask [advertise   not-advertise]</code> | Specifies an address range for which a single route will be advertised. |

This example shows how to configure route summarization between OSPF areas:

```
FWSM(config-router)# area 17 range 12.1.0.0 255.255.0.0
```

## Configuring Route Summarization when Redistributing Routes into OSPF

When routes from other protocols are redistributed into OSPF, each route is advertised individually in an external LSA. However, you can configure the Cisco IOS software to advertise a single route for all the redistributed routes that are covered by a specified network address and mask. This configuration decreases the size of the OSPF link-state database.

To configure the software advertisement on one summary route for all redistributed routes covered by a network address and mask, perform this task in router configuration mode:

| Command                                                                                                               | Purpose                                                                                                                                                                                    |
|-----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>FWSM(config-router)# <b>summary-address</b> {{ip-address mask}  {prefix mask}} [not-advertise] [tag tag]</code> | Specifies an address and mask that covers redistributed routes, so that only one summary route is advertised. Use the optional <b>not-advertise</b> keyword to filter out a set of routes. |

This example shows how to configure route summarization when redistributing routes into OSPF:

```
FWSM(config-router)# summary-address 12.1.0.0 255.255.0.0
```

## Creating Virtual Links

With OSPF all areas must be connected to a backbone area. If there is a break in backbone continuity, or the backbone is purposefully partitioned, you can establish a virtual link. The two end points of a virtual link are area border routers. The virtual link must be configured in both routers. The configuration information in each router consists of the other virtual end point (the other area border router) and the nonbackbone area that the two routers have in common (called the transit area). Virtual links cannot be configured through stub areas.

To establish a virtual link, follow this task in router configuration mode:

| Command                                                                                                                                                                                                                                                                                                                                                                                                                                      | Purpose                     |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|
| FWSM(config-router)# <b>area</b> <i>area-id</i> <b>virtual-link</b> <i>router-id</i> [ <b>authentication</b> [ <b>message-digest</b>   <b>null</b> ]] [ <b>hello-interval</b> <i>seconds</i> ] [ <b>retransmit-interval</b> <i>seconds</i> ] [ <b>transmit-delay</b> <i>seconds</i> ] [ <b>dead-interval</b> <i>seconds</i> ] [[ <b>authentication-key</b> <i>key</i> ]   [ <b>message-digest-key</b> <i>key-id</i> <b>md5</b> <i>key</i> ]] | Establishes a virtual link. |

This example shows how to create virtual links:

```
FWSM(config-router)# area 16 virtual-link 1.1.1.1
```

To display information about virtual links, use the **show ip ospf virtual-links** EXEC command.

To display the router ID of an OSPF router, use the **show ip ospf** EXEC command

## Generating a Default Route

You can force an autonomous system boundary router to generate a default route into an OSPF routing domain. Whenever you specifically configure redistribution of routes into an OSPF routing domain, the router automatically becomes an autonomous system boundary router. However, an autonomous system boundary router does not by default generate a default route into the OSPF routing domain.

To force the autonomous system boundary router to generate a default route, perform this task in router configuration mode:

| Command                                                                                                                                                                                         | Purpose                                                                                                |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| FWSM(config-router)# <b>default-information originate</b> [ <b>always</b> ] [ <b>metric</b> <i>metric-value</i> ] [ <b>metric-type</b> <i>type-value</i> ] [ <b>route-map</b> <i>map-name</i> ] | Forces the autonomous system boundary router to generate a default route into the OSPF routing domain. |

This example shows how to generate a default route:

```
FWSM(config-router)# default-information originate always
```

## Changing the OSPF Administrative Distances

An administrative distance is a rating of the trustworthiness of a routing information source, such as an individual router or a group of routers. An administrative distance numerically is an integer from 0 to 255. In general, the higher the value, the lower the trust rating. An administrative distance of 255 means the routing information source cannot be trusted and should be ignored.

OSPF uses three different administrative distances: intra-area, interarea, and external. Routes within an area are intra-area; routes to another area are interarea; and routes from another routing domain learned through redistribution are external. The default distance for each type of route is 110.

To change any of the OSPF distance values, perform this task in router configuration mode:

| Command                                                                                                  | Purpose                           |
|----------------------------------------------------------------------------------------------------------|-----------------------------------|
| <code>FWSM(config-router)# distance ospf {[intra-area dist1] [inter-area dist2] [external dist3]}</code> | Changes the OSPF distance values. |

This example shows how to change the OSPF administrative distance:

```
FWSM(config-router)# distance intra-ares 90 inter-area 95 external 100
```

## Configuring Route Calculation Timers

You can configure the delay time between when OSPF receives a topology change and when it starts a shortest path first (SPF) calculation. You also can configure the hold time between two consecutive SPF calculations. To configure the route calculation time, perform this task in router configuration mode:

| Command                                                             | Purpose                              |
|---------------------------------------------------------------------|--------------------------------------|
| <code>FWSM(config-router)# timers spf spf-delay spf-holdtime</code> | Configures route calculation timers. |

This example shows how to configure route calculation timers:

```
FWSM(config-router)# timers spf 10 120
```

## Logging Neighbors Going Up or Down

By default, the system sends a syslog message when an OSPF neighbor goes up or down.

Configure this command if you want to know about OSPF neighbors going up or down without turning on the **debug ip ospf adjacency EXEC** command. The **log-adj-changes router** configuration command provides a higher level view of the peer relationship with less output. Configure **log-adj-changes detail** if you want to see messages for each state change.

If you turned off this feature and want to restore it, perform this task in router configuration mode:

| Command                                                    | Purpose                                                     |
|------------------------------------------------------------|-------------------------------------------------------------|
| <code>FWSM(config-router)# log-adj-changes [detail]</code> | Sends syslog message when an OSPF neighbor goes up or down. |

This example shows how to log neighbors:

```
FWSM(config-router)# log-adj-changes detail
```

## Changing the LSA Group Pacing

The OSPF LSA group pacing feature allows the router to group OSPF LSAs and pace the refreshing, check summing, and aging functions. Group pacing results in more efficient use of the router.

The router groups OSPF LSAs and paces these functions so that sudden increases in CPU usage and network resources are avoided. This feature is most beneficial to large OSPF networks.

OSPF LSA group pacing is enabled by default. The default group pacing interval for refreshing, check summing, and aging usually is appropriate, and you need not configure this feature.

## Original LSA Behavior

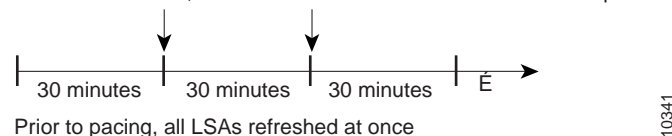
Each OSPF LSA has an age, which indicates whether the LSA is still valid. When the LSA reaches the maximum age (1 hour), it is discarded. During the aging process, the originating router sends a refresh packet every 30 minutes to refresh the LSA. Refresh packets are sent to keep the LSA from expiring, whether there has been a change in the network topology or not. Check summing is performed on all LSAs every 10 minutes. The router keeps track of LSAs it generates and LSAs it receives from other routers. The router refreshes LSAs it generated; it ages the LSAs it received from other routers.

Before the LSA group pacing feature was introduced, the Cisco IOS software would perform refreshing on a single timer, and check summing and aging on another timer. In the case of refreshing, for example, the software would scan the whole database every 30 minutes, refreshing every LSA the router generated, regardless of how old it was.

Figure 4-3 shows all the LSAs being refreshed at the same time. This process wasted CPU resources because only a small portion of the database needed to be refreshed. A large OSPF database (several thousand LSAs) might have thousands of LSAs with different ages. Refreshing on a single timer resulted in the age of all LSAs becoming synchronized, which resulted in increased CPU processing at once. A large number of LSAs might cause a sudden increase of network traffic, consuming a large amount of network resources in a short period of time.

**Figure 4-3** OSPF LSAs on a Single Timer Without Group Pacing

All LSAs refreshed, 120 external LSAs on Ethernet need three packets



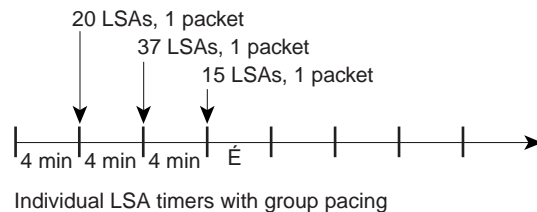
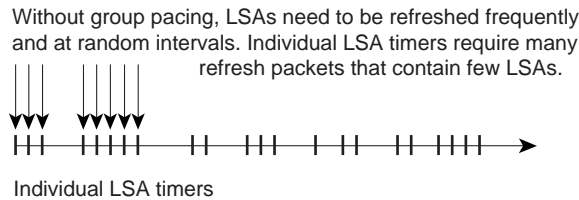
## LSA Group Pacing with Multiple Timers

This problem is solved by configuring each LSA to have its own timer. Each LSA gets refreshed when it is 30 minutes old, independent of other LSAs, so the CPU is used only when necessary. However, LSAs being refreshed at frequent, random intervals would require many packets for the few refreshed LSAs the router must send out, which would be inefficient use of bandwidth.

Therefore, the router delays the LSA refresh function for an interval of time instead of performing it when the individual timers are reached. The accumulated LSAs constitute a group, which is then refreshed and sent out in one packet or more. The refresh packets are paced as are the check summing and aging. The pacing interval is configurable; it defaults to 4 minutes, which is randomized to further avoid synchronization.

Figure 4-4 shows refresh packets. The first timeline shows individual LSA timers; the second timeline shows individual LSA timers with group pacing.

Figure 4-4 OSPF LSAs on Individual Timers with Group Pacing



10471

The group pacing interval is inversely proportional to the number of LSAs the router is refreshing, check summing, and aging. For example, if you have approximately 10,000 LSAs, decreasing the pacing interval would benefit you. If you have a very small database (40 to 100 LSAs), increasing the pacing interval to 10 to 20 minutes might benefit you slightly.

The default value of pacing between LSA groups is 240 seconds (4 minutes). The range is from 10 seconds to 1800 seconds (30 minutes). To change the LSA group pacing interval, perform this task in router configuration mode:

| Command                                                           | Purpose                           |
|-------------------------------------------------------------------|-----------------------------------|
| <code>FWSM(config-router)# timers lsa-group-pacing seconds</code> | Changes the group pacing of LSAs. |

The following example changes the OSPF pacing between LSA groups to 280 seconds:

```
FWSM(config-router)# timers lsa-group-pacing 280
FWSM(config-router)# interface inside
```

## Blocking OSPF LSA Flooding

By default, OSPF floods new LSAs over all interfaces in the same area, except the interface on which the LSA arrives. Some redundancy is desirable, because it ensures substantial flooding. However, too much redundancy can waste bandwidth and might destabilize the network due to excessive link and CPU usage in certain topologies, such as a fully meshed topology.

You can block OSPF flooding of LSAs two ways, depending on the type of networks:

- On broadcast, nonbroadcast, and point-to-point networks, you can block flooding over specified OSPF interfaces.
- On point-to-multipoint networks, you can block flooding to a specified neighbor.

On broadcast, nonbroadcast, and point-to-point networks, to prevent flooding of OSPF LSAs, perform this task in interface configuration mode:

| Command                                                           | Purpose                                                   |
|-------------------------------------------------------------------|-----------------------------------------------------------|
| <code>FWSM(config-if)# <b>ospf database-filter all out</b></code> | Blocks the flooding of OSPF LSA packets to the interface. |

On point-to-multipoint networks, to prevent flooding of OSPF LSAs, perform this task in router configuration mode:

| Command                                                                              | Purpose                                                            |
|--------------------------------------------------------------------------------------|--------------------------------------------------------------------|
| <code>FWSM(config-router)# <b>neighbor ip-address database-filter all out</b></code> | Blocks the flooding of OSPF LSA packets to the specified neighbor. |

## Ignoring MOSPF LSA Packets

Cisco routers do not support LSA type 6 Multicast OSPF (MOSPF). If the routers receive these packets, they generate syslog messages. If the router is receiving many MOSPF packets, you might want to configure the router to ignore the packets, which prevent a large number of syslog messages. To configure the router to ignore these packets, perform this task in router configuration mode:

| Command                                                   | Purpose                                                                                 |
|-----------------------------------------------------------|-----------------------------------------------------------------------------------------|
| <code>FWSM(config-router)# <b>ignore lsa mospf</b></code> | Prevents the router from generating syslog messages when it receives MOSPF LSA packets. |

The following example shows how to prevent flooding of OSPF LSAs to broadcast, nonbroadcast, or point-to-point networks reachable through Ethernet interface 0:

```
FWSM(config-router)# router ospf 2
FWSM(config-router)# ignore lsa mospf
FWSM(config-interface)# ospf database-filter all out
FWSM(config-interface)# router ospf 2
FWSM(config)# show ip ospf flood-list inside
```

```
Interface inside, Queue length 0
```

The following example shows how to prevent flooding of OSPF LSAs to point-to-multipoint networks to the neighbor at IP address 1.2.3.4:

```
FWSM(config-router)# router ospf 109
FWSM(config-router)# neighbor 1.2.3.4 database-filter all out
```

## Displaying OSPF Update Packet Pacing

The former OSPF implementation for sending update packets was not efficient. Some update packets were getting lost in situations where the link was slow, a neighbor could not receive the updates quickly enough, or the router was out of buffer space. For example, packets might be dropped if either of the following topologies existed:

- A fast router was connected to a slower router over a point-to-point link.
- During flooding, several neighbors sent updates to a single router at the same time.

OSPF update packets are now automatically paced so they are not sent less than 33 milliseconds apart. Pacing is also added between resends to increase efficiency and minimize lost retransmissions. You also can display the LSAs waiting to be sent out an interface. The benefit of the pacing is that OSPF update and retransmission packets are sent more efficiently.

There are no configuration tasks for this feature; it occurs automatically. To observe OSPF packet pacing by displaying a list of LSAs waiting to be flooded over a specified interface, perform this task in EXEC mode:

| Command                                                                      | Purpose                                                          |
|------------------------------------------------------------------------------|------------------------------------------------------------------|
| Router# <code>show ip ospf flood-list interface-type interface-number</code> | Displays a list of LSAs waiting to be flooded over an interface. |

## Area Border Router Type 3 LSA Filtering

The area border router Type 3 LSA filtering feature extends the capability of an area border router that is running the OSPF protocol to filter type 3 LSAs between different OSPF areas. This feature allows only specified prefixes to be sent from one area to another area and restricts all other prefixes. This type of area filtering can be applied out of a specific OSPF area, into a specific OSPF area, or into and out of the same OSPF areas at the same time. This feature is supported by the addition of the **area filter-list** command.

The OSPF ABR Type 3 LSA filtering feature provides improved control of route distribution between OSPF areas.

Only Type 3 LSAs that originate from an area border router are filtered.

## Configuring ABR Type 3 LSA Filtering

To filter interarea routes into a specified area, perform this task beginning in router configuration mode:

|        | Command                                                                                                                         | Purpose                                                                   |
|--------|---------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|
| Step 1 | <code>FWSM(config)#router ospf process-id</code>                                                                                | Enables OSPF routing, which places you in router configuration mode.      |
| Step 2 | <code>FWSM(config-router)#area area-id filter-list prefix name in</code>                                                        | Configures the router to filter interarea routes into the specified area. |
| Step 3 | <code>FWSM(config-router)#ip prefix-list list-name [seq seq-value] deny   permit network/len [ge ge-value] [le le-value]</code> | Creates a prefix list with the name specified for the list name argument. |

To filter interarea routes out of a specified area, perform the following task beginning in router configuration mode:

|        | Command                                                                                                                                    | Purpose                                                                     |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| Step 1 | <code>FWSM(config)#router ospf<br/>process-id</code>                                                                                       | Enables OSPF routing, which places you in router configuration mode.        |
| Step 2 | <code>FWSM(config-router)#area<br/>area-id filter-list prefix name<br/>out</code>                                                          | Configures the router to filter interarea routes out of the specified area. |
| Step 3 | <code>FWSM(config-router)#ip<br/>prefix-list name [seq<br/>seq-value] deny   permit<br/>network/len [ge ge-value] [le<br/>le-value]</code> | Creates a prefix list with the name specified for the list-name argument.   |

## Monitoring and Maintaining OSPF

You can display specific statistics such as the contents of IP routing tables, caches, and databases. Information provided can be used to determine resource utilization and solve network problems. You can also display information about node reachability and discover the routing path that your device packets are taking through the network.

To display various routing statistics, perform one of these tasks in EXEC mode, as needed:

| Command                                                                                                | Purpose                                                                                                         |
|--------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| <code>FWSM# show ip ospf [process-id]</code>                                                           | Displays general information about OSPF routing processes.                                                      |
| <code>FWSM# show ip ospf border-routers</code>                                                         | Displays the internal OSPF routing table entries to the area border router and autonomous system border router. |
| <code>FWSM# show ip ospf [process-id [area-id]] database</code>                                        |                                                                                                                 |
| <code>FWSM# show ip ospf [process-id [area-id]] database<br/>[database-summary]</code>                 |                                                                                                                 |
| <code>FWSM# show ip ospf [process-id [area-id]] database<br/>[router] [self-originate]</code>          |                                                                                                                 |
| <code>FWSM# show ip ospf [process-id [area-id]] database<br/>[router] [adv-router [ip-address]]</code> |                                                                                                                 |
| <code>FWSM# show ip ospf [process-id [area-id]] database<br/>[router] [link-state-id]</code>           |                                                                                                                 |
| <code>FWSM# show ip ospf [process-id [area-id]] database<br/>[network] [link-state-id]</code>          |                                                                                                                 |
| <code>FWSM# show ip ospf [process-id [area-id]] database<br/>[summary] [link-state-id]</code>          |                                                                                                                 |
| <code>FWSM# show ip ospf [process-id [area-id]] database<br/>[asbr-summary] [link-state-id]</code>     |                                                                                                                 |
| <code>FWSM# show ip ospf [process-id [area-id]] database<br/>[external] [link-state-id]</code>         |                                                                                                                 |
| <code>FWSM# show ip ospf [process-id [area-id]] database<br/>[nssa-external] [link-state-id]</code>    |                                                                                                                 |
| <code>FWSM# show ip ospf [process-id [area-id]] database<br/>[opaque-link] [link-state-id]</code>      |                                                                                                                 |
| <code>FWSM# show ip ospf [process-id [area-id]] database<br/>[opaque-area] [link-state-id]</code>      |                                                                                                                 |

| Command                                                                                         | Purpose                                                                                             |
|-------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| <code>FWSM# show ip ospf [process-id [area-id]] database [opaque-as] [link-state-id]</code>     | Displays lists of information related to the OSPF database.                                         |
| <code>FWSM# show ip ospf flood-list interface interface-type</code>                             | Displays a list of LSAs waiting to be flooded over an interface (to observe OSPF packet pacing).    |
| <code>FWSM# show ip ospf interface [interface-type interface-number]</code>                     | Displays OSPF-related interface information.                                                        |
| <code>FWSM# show ip ospf neighbor [interface-name] [neighbor-id] detail</code>                  | Displays OSPF neighbor information on a per-interface basis.                                        |
| <code>FWSM# show ip ospf request-list [neighbor] [interface] [interface-neighbor]</code>        | Displays a list of all LSAs requested by a router.                                                  |
| <code>FWSM# show ip ospf retransmission-list [neighbor] [interface] [interface-neighbor]</code> | Displays a list of all LSAs waiting to be resent.                                                   |
| <code>FWSM# show ip ospf [process-id] summary-address</code>                                    | Displays a list of all summary address redistribution information configured under an OSPF process. |
| <code>FWSM# show ip ospf virtual-links</code>                                                   | Displays OSPF-related virtual links information.                                                    |

To restart an OSPF process, perform this task in configuration mode:

| Command                                                                                                                          | Purpose                                                     |
|----------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| <code>FWSM(config)# clear ip ospf pid {process   redistribution   counters [neighbor [neighbor-interface] [neighbor-id]]}</code> | Clears redistribution based on the OSPF routing process ID. |

## Configuring IPSec for Management

Internet Protocol Security (IPSec) provides security for transmission of sensitive information over unprotected networks such as the Internet. IPSec operates at the network layer, protecting and authenticating IP packets between participating IPSec devices (peers), such as Firewall Services Modules.

IPSec provides the following optional network security services. A local security policy determines the use of one or more of these services:

- Data Confidentiality—The IPSec sender can encrypt packets before transmitting them across a network.
- Data Integrity—The IPSec receiver can authenticate packets sent by the IPSec sender to ensure that the data has not been altered during transmission.
- Data Origin Authentication—The IPSec receiver can authenticate the source of the IPSec packets sent. This service is dependent upon the data integrity service.
- Anti-Replay—The IPSec receiver can detect and reject replayed packets.



### Note

The term *data authentication* indicates data-integrity and data-origin authentication. Within this document, the term also includes antireplay services, unless otherwise specified.

IPSec provides controlled tunnels between two peers, such as two Firewall Services Modules. These tunnels are sets of security associations that are established between two remote IPSec peers (modules). You define which packets are considered sensitive and should be sent through these controlled tunnels,

and you define the parameters that should be used to protect these sensitive packets by specifying the characteristics of these tunnels. When the IPSec peer sees a sensitive packet, it sets up the appropriate controlled tunnel and sends the packet through the tunnel to the remote peer.

For detailed information about IPSec, refer to the following URL:

[http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix\\_60/ipsec/index.htm](http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_60/ipsec/index.htm)

The following steps describe a minimal IPSec configuration where the IPSec security associations are established through Internet Key Exchange (IKE).

To configure IPSec with IKE for the module, perform this task:

|         | Command                                                                                                                                                                      | Purpose                                                                                                                                                                                                                                    |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1  | <code>FWSM(config)# <b>access-list</b> access-list-module<br/>{deny   permit} ip source source-netmask<br/>destination destination-netmask</code>                            | Creates an access list to define the traffic to protect.                                                                                                                                                                                   |
| Step 2  | <code>FWSM(config)# <b>crypto ipsec transform-set</b><br/>transform-set-module transform1 [transform2,<br/>transform3]</code>                                                | Configures 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 in <a href="#">Step 6</a> .               |
| Step 3  | <code>FWSM(config)# <b>crypto map</b> map-module seq-num<br/><b>ipsec-isakmp</b></code>                                                                                      | Creates a crypto map entry in IPSec ISAKMP mode.                                                                                                                                                                                           |
| Step 4  | <code>FWSM(config)# <b>crypto map</b> map-module seq-num <b>match</b><br/><b>address</b> access-list-module</code>                                                           | Assigns an access list to a crypto map entry.                                                                                                                                                                                              |
| Step 5  | <code>FWSM(config)# <b>crypto map</b> map-module seq-num <b>set</b><br/><b>peer</b> ip-address</code>                                                                        | Specifies the peer to which the IPSec-protected traffic can be forwarded. The security association is set up with the peer having an IP address of 192.168.1.100. Specify multiple peers by repeating this command.                        |
| Step 6  | <code>FWSM(config)# <b>crypto map</b> map-module seq-num <b>set</b><br/><b>transform-set</b> transform-set-module1<br/>[transform-set-module2, transform-set-module6]</code> | Specifies which transform sets are allowed for this crypto map entry. Lists multiple transform sets in order of priority (highest priority first). You can specify up to six transform sets.                                               |
| Step 7  | <code>FWSM(config)# <b>crypto map</b> map-module seq-num <b>set</b><br/><b>security-association lifetime</b> {seconds seconds  <br/>kilobytes kilobytes}</code>              | (Optional) Specifies a 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. |
| Step 8  | <code>FWSM(config)# <b>crypto map</b> map-module seq-num <b>set</b><br/><b>pfs</b> [group1   group2]</code>                                                                  | (Optional) Specifies that IPSec should require perfect forward secrecy (PFS) when requesting new security associations for this crypto map entry, or should require PFS in requests received from the peer.                                |
| Step 9  | <code>FWSM(config)# <b>crypto dynamic-map</b><br/>dynamic-map-module dynamic-seq-num <b>match</b> <b>address</b><br/>access-list-module</code>                               | (Optional) Assigns an access list to a dynamic crypto map entry, which determines which traffic should be protected and which traffic should not be protected.                                                                             |
| Step 10 | <code>FWSM(config)# <b>crypto dynamic-map</b><br/>dynamic-map-module dynamic-seq-num <b>set</b> <b>peer</b><br/>ip-address</code>                                            | (Optional) Specifies the peer to which the IPSec-protected traffic can be forwarded. This is rarely configured in dynamic crypto map entries because dynamic crypto map entries are often used for unknown peers.                          |

|         | Command                                                                                                                                                                              | Purpose                                                                                                                                                                                                                                            |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 11 | <code>FWSM(config)# crypto dynamic-map<br/>dynamic-map-module dynamic-seq-num set<br/>transform-set transform-set-module1,<br/>[transform-set-module2, transform-set-module9]</code> | Specifies which transform sets are allowed for this dynamic crypto map entry. Lists multiple transform sets in order of priority (highest priority first).                                                                                         |
| Step 12 | <code>FWSM(config)# crypto dynamic-map<br/>dynamic-map-module dynamic-seq-num set<br/>security-association lifetime {seconds seconds  <br/>kilobytes kilobytes}</code>               | (Optional) Specifies a security association lifetime for the dynamic 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. |
| Step 13 | <code>FWSM(config)# crypto dynamic-map<br/>dynamic-map-module dynamic-seq-num set pfs<br/>[group1   group2]</code>                                                                   | (Optional) Specifies that IPSec should request PFS when requesting new security associations for this dynamic crypto map entry, or should demand PFS in requests received from the peer.                                                           |
| Step 14 | <code>FWSM(config)# crypto map map-module seq-num<br/>ipsec-isakmp dynamic dynamic-map-module</code>                                                                                 | Adds 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.                                     |
| Step 15 | <code>FWSM(config)# crypto map map-module interface<br/>interface-module</code>                                                                                                      | Applies a crypto map set to an interface on which the IPSec traffic will be evaluated.                                                                                                                                                             |
| Step 16 | <code>FWSM# sysopt connection permit-ipsec</code>                                                                                                                                    | Specifies that IPSec traffic be implicitly trusted (permitted).                                                                                                                                                                                    |

In the Firewall Services Module, VPN and IPSec are available only for management purposes. You cannot establish IPSec tunnels across the firewall; any tunnel initiated by a VPN client on another switch should terminate at the Firewall Services Module. The CLI commands you use to configure IPSec for management have not changed from PIX except for those listed in [Table A-6 on page A-5](#). Refer to the PIX documentation for details about configuring IPSec.