Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Configuration Guide

Release 2.2

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Text Part Number: OL-5891-01

About This Guide

This guide contains the following sections:

Document Objectives

The purpose of this guide is to help you configure the Firewall Services Module (FWSM) for the most common scenarios using the command line interface. It does not cover every feature, but describes those tasks most commonly required for configuration.

Audience

This guide is for network managers who perform any of the following tasks:

- Managing network security
- Installing and configuring firewalls
- Managing default and static routes, and TCP and UDP services

Related Documentation

For more information, refer to the following documentation set for the FWSM:

- *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*
- *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module System Messages Guide*
- *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Installation Note*
- *Release Notes for the Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module*

Document Organization

This guide includes the following chapters and appendixes:

- "Quick Start Steps" provides pointers to the minimum configuration required for routed or transparent mode.
- Chapter 1, "Introduction to the Firewall Services Module," describes the system requirements and features.
- Chapter 2, "Configuring the Switch for the Firewall Services Module," tells how to configure the switch for use with the FWSM.
- Chapter 3, "Connecting to the Firewall Services Module and Managing the Configuration," tells how to access the FWSM command line interface (CLI) and manage the configuration.
- Chapter 4, "Configuring the Firewall Mode," tells how to set the firewall mode.
- Chapter 5, "Managing Security Contexts," tells how to configure multiple security contexts.
- Chapter 6, "Configuring Basic Settings," tells how to configure basic settings that are either essential or useful to the operation of your FWSM.
- Chapter 7, "Configuring Bridging Parameters and ARP Inspection," tells how to customize the operation of the transparent firewall.
- Chapter 8, "Configuring IP Addresses, Routing, and DHCP," tells how to configure IP addresses, static routes, dynamic routing, and DHCP.
- Chapter 9, "Configuring Network Address Translation," tells how to configure Network Address Translation (NAT).
- Chapter 10, "Controlling Network Access with Access Control Lists," tells how to control network access through the FWSM using access control lists (ACLs).
- Chapter 11, "Allowing Remote Management," tells how to allow remote management access to the FWSM.
- Chapter 12, "Configuring AAA," tells how to configure AAA, which includes command authorization, CLI access authentication, and AAA for traffic through the FWSM.
- Chapter 13, "Configuring Application Protocol Inspection," tells how to configure inspection engines.
- Chapter 14, "Filtering HTTP, HTTPS, or FTP Requests Using an External Server," tells how to configure filtering.

- Chapter 15, “Using Failover,” tells how to configure a primary and secondary FWSM for redundancy.
- Chapter 16, “Managing Software and Configuration Files,” tells how to upgrade or reinstall FWSM software.
- Chapter 17, “Monitoring and Troubleshooting the Firewall Services Module,” tells how to monitor and troubleshoot the FWSM. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module System Messages Guide* for detailed information about system logging.
- Appendix A, “Specifications,” lists the specifications for the FWSM.
- Appendix B, “Sample Configurations,” shows some common scenarios and the configurations that support them.
- Appendix C, “Understanding the Command-Line Interface,” describes the CLI.
- Appendix D, “Addresses, Protocols, and Ports Reference,” provides reference information, including lists of TCP, UDP, and ICMP port types, and common subnet masks.
- Appendix E, “Acronyms and Abbreviations,” lists acronyms and abbreviations used in this guide.
- “Index” provides easy access to topics within the guide.

Document Conventions

This guide uses the following conventions:

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Syntax formatting is described in the “Syntax Formatting” section on page C-2.

Obtaining Documentation

Cisco documentation and additional literature are available on Cisco.com. Cisco also provides several ways to obtain technical assistance and other technical resources. These sections explain how to obtain technical information from Cisco Systems.

Cisco.com

You can access the most current Cisco documentation at this URL:

http://www.cisco.com/univercd/home/home.htm

You can access the Cisco website at this URL:

http://www.cisco.com

You can access international Cisco websites at this URL:

http://www.cisco.com/public/countries_languages.shtml

Ordering Documentation

You can find instructions for ordering documentation at this URL:

http://www.cisco.com/univercd/cc/td/doc/es_inpck/pdi.htm

You can order Cisco documentation in these ways:

- Registered Cisco.com users (Cisco direct customers) can order Cisco product documentation from the Ordering tool:

  http://www.cisco.com/en/US/partner/ordering/index.shtml

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco Systems Corporate Headquarters (California, USA) at 408 526-7208 or, elsewhere in North America, by calling 800 553-NETS (6387).

Documentation Feedback

You can send comments about technical documentation to bug-doc@cisco.com.

You can submit comments by using the response card (if present) behind the front cover of your document or by writing to the following address:

Cisco Systems
Attn: Customer Document Ordering
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Obtaining Technical Assistance

For all customers, partners, resellers, and distributors who hold valid Cisco service contracts, Cisco Technical Support provides 24-hour-a-day, award-winning technical assistance. The Cisco Technical Support Website on Cisco.com features extensive online support resources. In addition, Cisco Technical Assistance Center (TAC) engineers provide telephone support. If you do not hold a valid Cisco service contract, contact your reseller.

Cisco Technical Support Website

The Cisco Technical Support Website provides online documents and tools for troubleshooting and resolving technical issues with Cisco products and technologies. The website is available 24 hours a day, 365 days a year at this URL:

http://www.cisco.com/techsupport

Access to all tools on the Cisco Technical Support Website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register at this URL:

http://tools.cisco.com/RPF/register/register.do

Submitting a Service Request

Using the online TAC Service Request Tool is the fastest way to open S3 and S4 service requests. (S3 and S4 service requests are those in which your network is minimally impaired or for which you require product information.) After you describe your situation, the TAC Service Request Tool automatically provides recommended solutions. If your issue is not resolved using the recommended resources, your service request will be assigned to a Cisco TAC engineer. The TAC Service Request Tool is located at this URL:

http://www.cisco.com/techsupport/servicerequest

For S1 or S2 service requests or if you do not have Internet access, contact the Cisco TAC by telephone. (S1 or S2 service requests are those in which your production network is down or severely degraded.) Cisco TAC engineers are assigned immediately to S1 and S2 service requests to help keep your business operations running smoothly.

To open a service request by telephone, use one of the following numbers:

Asia-Pacific: +61 2 8446 7411 (Australia: 1 800 805 227)
EMEA: +32 2 704 55 55
USA: 1 800 553 2447

For a complete list of Cisco TAC contacts, go to this URL:

http://www.cisco.com/techsupport/contacts

Definitions of Service Request Severity

To ensure that all service requests are reported in a standard format, Cisco has established severity definitions.

Severity 1 (S1)—Your network is “down,” or there is a critical impact to your business operations. You and Cisco will commit all necessary resources around the clock to resolve the situation.

Severity 2 (S2)—Operation of an existing network is severely degraded, or significant aspects of your business operation are negatively affected by inadequate performance of Cisco products. You and Cisco will commit full-time resources during normal business hours to resolve the situation.

Severity 3 (S3)—Operational performance of your network is impaired, but most business operations remain functional. You and Cisco will commit resources during normal business hours to restore service to satisfactory levels.

Severity 4 (S4)—You require information or assistance with Cisco product capabilities, installation, or configuration. There is little or no effect on your business operations.

Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

- Cisco Marketplace provides a variety of Cisco books, reference guides, and logo merchandise. Visit Cisco Marketplace, the company store, at this URL:

  http://www.cisco.com/go/marketplace/

- The Cisco *Product Catalog* describes the networking products offered by Cisco Systems, as well as ordering and customer support services. Access the Cisco Product Catalog at this URL:

  http://cisco.com/univercd/cc/td/doc/pcat/

- *Cisco Press* publishes a wide range of general networking, training and certification titles. Both new and experienced users will benefit from these publications. For current Cisco Press titles and other information, go to Cisco Press at this URL:

  http://www.ciscopress.com

- *Packet* magazine is the Cisco Systems technical user magazine for maximizing Internet and networking investments. Each quarter, Packet delivers coverage of the latest industry trends, technology breakthroughs, and Cisco products and solutions, as well as network deployment and troubleshooting tips, configuration examples, customer case studies, certification and training information, and links to scores of in-depth online resources. You can access Packet magazine at this URL:

  http://www.cisco.com/packet

- *iQ Magazine* is the quarterly publication from Cisco Systems designed to help growing companies learn how they can use technology to increase revenue, streamline their business, and expand services. The publication identifies the challenges facing these companies and the technologies to help solve them, using real-world case studies and business strategies to help readers make sound technology investment decisions. You can access iQ Magazine at this URL:

  http://www.cisco.com/go/iqmagazine

- *Internet Protocol Journal* is a quarterly journal published by Cisco Systems for engineering professionals involved in designing, developing, and operating public and private internets and intranets. You can access the Internet Protocol Journal at this URL:

  http://www.cisco.com/ipj

- World-class networking training is available from Cisco. You can view current offerings at this URL:

  http://www.cisco.com/en/US/learning/index.html

Quick Start Steps

The following sections describe the minimum configuration required for the Firewall Services Module (FWSM) in routed mode or transparent mode:

Routed Firewall Configuration Steps

Follow these steps to configure the FWSM in routed mode:

	Task	Description
Step 1	Assigning VLANs to the Firewall Services Module, page 2-2	On the switch, you need to assign VLANs to the FWSM so the FWSM can send and receive traffic on the switch.
Step 2	(Might be required) Adding Switched Virtual Interfaces to the MSFC, page 2-5	If you want the Multilayer Switch Feature Card (MSFC) to route between VLANs that are assigned to the FWSM, complete this procedure.
Step 3	Sessioning and Logging into the Firewall Services Module, page 3-1	From the switch CLI, you can session into the FWSM to access the FWSM CLI.
Step 4	(Might be required; multiple context mode only) Enabling or Disabling Multiple Context Mode, page 5-10	If you want to use multiple context mode and your FWSM is not already configured for it, or if you want to change back to single mode, follow this procedure.
Step 5	(Multiple context mode only) Configuring a Security Context, page 5-17	Add a security context.
Step 6	(Multiple context mode only) Changing Between Contexts and the System Execution Space, page 5-20	You must configure some settings in the system execution space, and some settings within the context, so you need to know how to switch between contexts and the system execution space.
Step 7	Setting the Name and Security Level, page 6-7	For each VLAN interface, you must set a name (such as inside or outside) and a security level.
Step 8	Assigning IP Addresses to Interfaces for a Routed Firewall, page 8-2	Assign an IP address to each interface.
Step 9	Configuring the Default Route, page 8-2	Create a default route to an upstream router.

	Task	Description
Step 10	Configure routing using one of these methods: • Configuring Static Routes, page 8-3 • (Single context mode only) Configuring OSPF, page 8-4 • (Single context mode only) Configuring RIP, page 8-18	In multiple context mode, static routing is the only routing method supported. In single mode, you have a choice of static, RIP, or OSPF. RIP support is for passive mode only.
Step 11	Use one or more of these NAT methods: • Using Dynamic NAT and PAT, page 9-15 • Using Static NAT, page 9-25 • Using Static PAT, page 9-26 • Bypassing NAT, page 9-28	You must specifically configure some interfaces to either use or bypass NAT. Typically, if you want to allow inside users to access the outside or other networks attached to the FWSM, configure dynamic NAT or PAT according to the "Using Dynamic NAT and PAT" section on page 15. If you want to allow outside users to access an inside host, then configure static NAT according to the "Using Static NAT" section on page 25. The FWSM offers a large amount of flexibility in your NAT configuration.
Step 12	Adding an Extended Access Control List, page 10-13	Before any traffic can go through the FWSM, you must create an ACL that permits traffic, and then apply it to an interface.

Transparent Firewall Configuration Steps

Follow these steps to configure the FWSM in transparent mode:

	Task	Description
Step 1	Assigning VLANs to the Firewall Services Module, page 2-2	On the switch, you need to assign VLANs to the FWSM so the FWSM can send and receive traffic on the switch.
Step 2	(Might be required) Adding Switched Virtual Interfaces to the MSFC, page 2-5	If you want the MSFC to route between VLANs that are assigned to the FWSM, complete this procedure.
Step 3	Sessioning and Logging into the Firewall Services Module, page 3-1	From the switch CLI, you can session into the FWSM to access the FWSM CLI.
Step 4	Setting the Firewall Mode, page 4-16	Before you configure any settings, you must set the firewall mode to transparent mode. Changing the mode clears your configuration.
Step 5	(Might be required; multiple context mode only) Enabling or Disabling Multiple Context Mode, page 5-10	If you want to use multiple context mode and your FWSM is not already configured for it, or if you want to change back to single mode, follow this procedure.
Step 6	(Multiple context mode only) Configuring a Security Context, page 5-17	Add a security context.
Step 7	(Multiple context mode only) Changing Between Contexts and the System Execution Space, page 5-20	You must configure some settings in the system execution space, and some settings within the context, so you need to know how to switch between contexts and the system execution space.
Step 8	Setting the Name and Security Level, page 6-7	For each VLAN interface, you must set a name (such as inside or outside) and a security level.
Step 9	Setting the Management IP Address for a Transparent Firewall, page 8-2	The transparent firewall requires a management IP address.
Step 10	Adding an Extended Access Control List, page 10-13	Before any traffic can go through the FWSM, you must create an ACL that permits traffic, and then apply it to an interface.

Introduction to the Firewall Services Module

The Firewall Services Module (FWSM) is a high-performance, space-saving, stateful firewall module that installs in the Catalyst 6500 series switches and the Cisco 7600 series routers.

Firewalls protect inside networks from unauthorized access by users on an outside network. The firewall can also protect inside networks from each other, for example, by keeping a human resources network separate from a user network. If you have network resources that need to be available to an outside user, such as a web or FTP server, you can place these resources on a separate network behind the firewall, called a *demilitarized zone* (DMZ). The firewall allows limited access to the DMZ, but because the DMZ only includes the public servers, an attack there only affects the servers and does not affect the other inside networks. You can also control when inside users access outside networks (for example, access to the Internet), by allowing only certain addresses out, by requiring authentication or authorization, or by coordinating with an external URL filtering server.

Note When discussing networks connected to a firewall, the *outside* network is in front of the firewall, the *inside* network is protected and behind the firewall, and a *DMZ*, while behind the firewall, allows limited access to outside users. Because the FWSM lets you configure many interfaces with varied security policies, including many inside interfaces, many DMZs, and even many outside interfaces if desired, these terms are used in a general sense only.

The FWSM includes many advanced features, such as multiple security contexts (similar to virtualized firewalls), transparent (Layer 2) firewall or routed (Layer 3) firewall operation, hundreds of interfaces, and many more features.

This chapter contains the following sections:

Chassis System Requirements

The switch models that support the FWSM include the following platforms:

- Catalyst 6500 series switches, with the following required components:
  - Supervisor engine with Cisco IOS software (known as supervisor IOS) *or* Catalyst operating system (OS). See Table 1-1 for supported supervisor engine and software releases.
  - Multilayer Switch Feature Card (MSFC 2) with Cisco IOS software. See Table 1-1 for supported Cisco IOS releases.
- Cisco 7600 series routers, with the following required components:
  - Supervisor engine with Cisco IOS software. See Table 1-1 for supported supervisor engine and software releases.
  - MSFC 2 with Cisco IOS software. See Table 1-1 for supported Cisco IOS releases.

Note The FWSM does not support a direct connection to a switch WAN port because WAN ports do not use static virtual local area networks (VLANs). However, the WAN port can connect to the MSFC, which can connect to the FWSM.

Table 1-1 shows the supervisor engine version, software, and supported FWSM features.

Table 1-1 Support for FWSM 2.2 Features

		FWSM Features:	
	Supervisor Engines[1]	**Multiple SVIs[2]**	**Transparent Firewall with Failover[3]**
Cisco IOS			
12.1(13)E	2	No	No
12.1(19)E	2	**Yes**	No
12.1(22)E and higher	2	**Yes**	**Yes**
12.2(14)SY and higher	2	**Yes**	No
12.2(14)SX	2, 720	No	No
12.2(17a)SX3	2, 720	**Yes**	**Yes**
12.2(17b)SXA	2, 720	**Yes**	**Yes**
12.2(17d)SXB and higher	2, 720	**Yes**	**Yes**
Catalyst OS[4]			
7.5(x)	2	No	No
7.6(1) through 7.6(4)	2	**Yes**	No
7.6(5) and higher	2	**Yes**	**Yes**
8.2(x)	2, 720	**Yes**	**Yes**
8.3(x)	2, 720	**Yes**	**Yes**

1. The FWSM does not support the supervisor 1 or 1A.
2. Supports multiple switched VLAN interfaces (SVIs) between the MSFC and FWSM. An SVI is a VLAN interface that is routed on the MSFC.
3. Supports transparent firewall mode when you use failover. Failover requires BPDU forwarding to the FWSM, or else you can have a loop. Other releases that do not support BPDU forwarding only support transparent mode without failover.
4. When you use Catalyst OS on the supervisor, you can use any of the supported Cisco IOS releases above on the MSFC. (When you use Cisco IOS software on the supervisor, you use the same release on the MSFC.) The supervisor software determines the FWSM feature support. For example, if you use Catalyst OS Release 7.6(1) on the supervisor and Cisco IOS 12.1(13)E on the MSFC, then the switch does support multiple SVIs, because Catalyst OS Release 7.6(1) supports multiple SVIs.

Features

This section describes the FWSM features, and includes the following topics:

General Features

Table 1-2 lists the features of the FWSM.

Table 1-2 General FWSM Features

Feature	Description
Transparent firewall or routed firewall mode	The firewall can run in one of the following modes: • Routed—The FWSM is considered to be a router hop in the network. It performs NAT[1] between connected networks. In single context mode, you can use OSPF[2] or passive RIP[3]. • Transparent—The FWSM acts like a "bump in the wire," and is not a router hop. The FWSM connects the same network on its inside and outside interfaces, but each interface must be on a different VLAN. No dynamic routing protocols or NAT are required.
Multiple security contexts	In multiple context mode, you can create up to 100 separate security contexts (depending on your software license). A security context is a virtual firewall that has its own security policy and interfaces. Multiple contexts are similar to having multiple stand-alone firewalls. Contexts are conveniently contained within a single module. You can run all security contexts in routed mode or in transparent mode; you cannot run some contexts in one mode and others in another. With the default software license, you can run up to two security contexts in addition to an admin context. For more contexts, you must purchase a license.
Resource management for security contexts	You can limit resources per context so one context does not use up too many resources. You create classes that define resource limitations as an absolute value or as a percentage, and then assign a context to one of these classes.
Communication between same security level	You can configure interfaces on the same security level to communicate with each other. This feature is off by default, and you can enable or disable this feature on a per context basis. In earlier releases, no communication between interfaces with the same security level was possible.
Bidirectional NAT and policy NAT	You can perform NAT on inside and outside addresses. For policy NAT, you can identify addresses to be translated using an extended ACL[4], which allows you more control in determining which addresses to translate.

Table 1-2 General FWSM Features (continued)

Feature	Description
Several ACL types	The FWSM supports the following ACLs: • Extended ACL to control IP traffic on an interface: – Inbound – Outbound • For transparent firewall mode, EtherType ACL to control non-IP traffic on an interface: – Inbound – Outbound • Standard ACL for OSPF route redistribution.
URL Filtering	Filter HTTP, HTTPS, and FTP requests using Websense Enterprise or Sentian by N2H2.
Dynamic Routing Protocols	Single context mode only. • RIP v1 and v2 (passive mode) • OSPF **Note** Multiple context mode supports static routing only.
DHCP server and DHCP relay	The FWSM acts as a DHCP[5] server. The FWSM also supports DHCP relay to forward DHCP requests to an upstream router.
Management	The FWSM supports the following management methods: • Cisco PDM for FWSM—Release 4.0 supports FWSM Release 2.2 features. PDM is a browser-based configuration tool that resides on the FWSM. The system administrator can configure multiple security contexts. If desired, individual context administrators can configure only their contexts. • Cisco Firewall MC[6]—Release 1.3.1 supports FWSM Release 2.2 features. For multiple context mode, Release 1.3.1 supports management of each context separately but does not support system-level operations, such as adding or deleting contexts, or the provisioning of failover in multiple mode. • CLI[7]
Login banners	You can define a text message to display when users log into the FWSM.
System message enhancements	You can configure ACLs to generate system messages when they match traffic. You can also set the level for a system message.

1. Network Address Translation
2. Open Shortest Path First
3. Routing Information Protocol
4. access control lists
5. Dynamic Host Configuration Protocol
6. Firewall Management Center
7. command-line interface

Stateful Inspection Feature

All traffic that goes through the firewall is inspected using the Adaptive Security Algorithm (ASA) and either allowed through or dropped. A simple packet filter can check for the correct source address, destination address, and ports, but it does not check that the packet sequence or flags are correct. A filter also checks every packet against the filter, which can be a slow process.

A stateful firewall like the FWSM, however, takes into consideration the state of a packet:

- Is this a new connection?

  If it is a new connection, the firewall has to check the packet against ACLs and perform other tasks to determine if the packet is allowed or denied. To perform this check, the first packet of the session goes through the "session management path," and depending on the type of traffic, it might also pass through the "control plane path."

  The session management path is responsible for the following tasks:

  - Performing the ACL checks
  - Performing route lookups
  - Allocating NAT translations (xlates)
  - Establishing sessions in the "fast path"

  Some packets that require Layer 7 inspection (the packet payload must be inspected or altered) are passed on to the control plane path. Layer 7 inspection engines are required for protocols that have two or more channels: a data channel, which uses well-known port numbers, and a control channel, which uses different port numbers for each session. These protocols include FTP, H.323, and SNMP.

Note The FWSM performs session management path and fast path processing on three specialized networking processors (NPs). The control plane path processing is performed in a general-purpose processor that also handles traffic directed to the FWSM and configuration and management tasks.

- Is this an established connection?

  If the connection is already established, the firewall does not need to re-check packets; most matching packets can go through the fast path in both directions. The fast path is responsible for the following tasks:

  - IP checksum verification
  - Session lookup
  - TCP sequence number check
  - NAT translations based on existing sessions
  - Layer 3 and Layer 4 header adjustments

  The following types of traffic go through the fast path:

  - Established TCP or UDP connections

    For UDP, which does not have sessions, the FWSM creates UDP connection state information so that it can also use the fast path.

  - ICMP control packets
  - Data packets for protocols that require Layer 7 inspection

  Some established session packets must continue to go through the session management path or the control plane path. Packets that go through the session management path include HTTP packets that require inspection or content filtering. Packets that go through the control plane path include the control packets for protocols that require Layer 7 inspection.

Other Protection Features

Table 1-3 describes the protection features provided by the FWSM. These features control network activity associated with specific kinds of attacks.

Table 1-3 Protection Features

Protection Feature	Description
ARP Inspection	For transparent firewall mode, you can enable ARP inspection. By default, ARP inspection is disabled on all interfaces; all ARP packets are allowed through the FWSM. When you enable ARP inspection, the FWSM compares the MAC address and IP address in all ARP packets to static entries in the ARP table. Enable this feature using the **arp inspection** command.
DNS Guard	DNS Guard identifies each outbound DNS[1] resolve request, and allows only a single DNS response. A host might query several servers for a response (in the case that the first server is slow in responding), but only the first answer to the request is allowed. All additional responses to the request are dropped by the firewall. This feature is always enabled. This feature is unrelated to the DNS inspection engine.
Flood Guard	Flood Guard controls the tolerance of the AAA server for unanswered login attempts. This helps to prevent a DoS[2] attack on AAA services in particular. This feature optimizes AAA system use. Flood Guard is enabled by default and can be controlled with the **floodguard** command.
Frag Guard	Frag Guard provides IP fragment protection, and can be configured with the **fragment** command. **Note** In FWSM 1.1, the default fragment size was 1, which caused the FWSM to drop all fragments by default. In FWSM 2.2, the default fragment size is 200 (the same as the PIX default).
ICMP Filtering	The FWSM automatically denies ICMP access to FWSM interfaces. This feature shields FWSM interfaces from detection by users on an external network. You can allow ICMP to FWSM interfaces using the **icmp** command.
Mail Guard	Mail Guard provides safe access for SMTP[3] connections from the outside to an inside messaging server. This feature lets you deploy a single mail server within the internal network without it being exposed to known security problems with some SMTP server implementations. This eliminates the need for an external mail relay (or bastion host) system. Mail Guard enforces a safe minimal set of SMTP commands to avoid an SMTP server system from being compromised. Enable this feature using the **fixup protocol smtp 25** command.
TCP Intercept	TCP Intercept protects inside systems from a DoS attack perpetrated by flooding an interface with TCP SYN[4] packets. Enable this feature by setting the maximum embryonic connections option of the **nat** and **static** commands. When the embryonic limit has been surpassed, the TCP intercept feature intercepts TCP SYN packets from clients to servers on a higher security level. The software establishes a connection with the client on behalf of the destination server and, if successful, establishes the connection with the server on behalf of the client, then combines the two half-connections together transparently. Thus, connection attempts from unreachable hosts will never reach the server. **Note** The PIX firewall accomplishes TCP intercept functionality using SYN cookies; the FWSM uses a different method, but accomplishes the same goal.
Unicast Reverse Path Forwarding	Unicast RPF helps mitigate problems caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP packets that lack a verifiable IP source address. Enable this feature using the **ip verify reverse-path** command.

1. Domain Name System
2. denial of service
3. Simple Mail Transfer Protocol
4. synchronization

How the Firewall Services Module Works

This section describes the network firewall functionality provided by the FWSM. It includes the following topics:

Security Policy Overview

A security policy determines which traffic is allowed to pass through the firewall to access another network. By default, no traffic can pass through the firewall. By applying ACLs to interfaces, you can determine which IP addresses and traffic types can pass through the interfaces to access other networks.

Note By default, the Cisco PIX firewall allows traffic to flow freely from an inside network (higher security level) to an outside network (lower security level). However, the FWSM does not allow *any* traffic to pass between interfaces unless you explicitly permit it with an ACL. This rule is true for both routed firewall mode and transparent firewall mode. While you still specify the security level for an interface on the FWSM, the security level does not provide explicit permission for traffic to travel from a high security interface to a low security interface. See the "Configuring Interfaces" section on page 6-6 for more information about how security levels work.

For routed firewall mode, in addition to ACLs, you can use Network Address Translation (NAT) between networks to further protect the real IP addresses of hosts.

If you have an AAA server, you can also apply AAA rules to users to control their access.

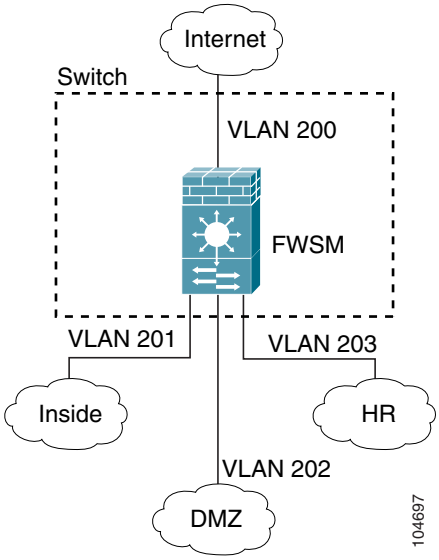
All of these features plus others, such as filters or inspection engines, make up the security policy of the firewall.

VLAN Interfaces

The FWSM does not include any external physical interfaces. Instead, it uses internal VLAN interfaces. For example, you assign VLAN 201 to the FWSM inside interface, and VLAN 200 to the outside interface. You assign these VLANs to physical switch ports, and hosts connect to those ports. When communication occurs between VLANs 201 and 200, the FWSM is the only available path between the VLANs, forcing traffic to be statefully inspected.

Figure 1-1 shows the FWSM with 4 interfaces: one outside interface (VLAN 200), one DMZ interface (VLAN 202), and two inside interfaces (VLAN 201 and 203).

Figure 1-1 VLAN Interfaces



How the Firewall Services Module Works with the Switch

You can install the FWSM in the Catalyst 6500 series switches and the Cisco 7600 series routers. The configuration of both series is identical, except for the following variations:

- The Catalyst 6500 series switches supports two software modes:
  - Cisco IOS software on both the switch supervisor and the integrated MSFC (known as "supervisor IOS").
  - Catalyst Operating System (OS) on the supervisor, and Cisco IOS software on the MSFC.

  For commands and configuration that are performed on the switch itself, both modes are described.
- The Cisco 7600 series routers support only Cisco IOS software.

Both series are referred to generically in this guide as the "switch."

The FWSM runs its own operating system, based on the PIX operating system. Although the PIX OS is similar to the FWSM OS, there are a number of differences. Many of the differences are enhancements that take advantage of the FWSM hardware and architecture.

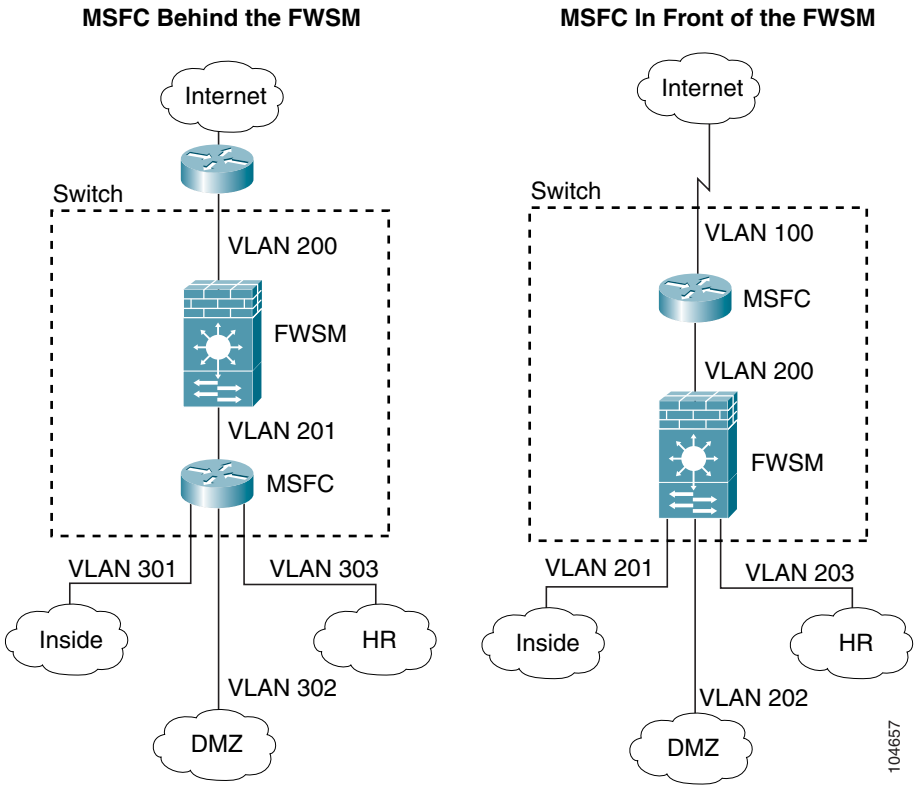
Using the MSFC

The switch includes a switching processor (the supervisor) and a router (the MSFC). Although you need the MSFC as part of your system, you do not have to use it. If you choose to do so, you can assign one or more VLAN interfaces to the MSFC (if your switch software version supports multiple SVIs; see Table 1-1 on page 1-2). In single context mode, you can place the MSFC in front of the firewall or behind the firewall (see Figure 1-2).

The location of the MSFC depends entirely on the VLANs that you assign to it. For example, the MSFC is behind the firewall in the example shown on the left side of Figure 1-2 because you assigned VLAN 201 to the inside interface of the FWSM. The MSFC is in front of the firewall in the example shown on the right side of Figure 1-2 because you assigned VLAN 200 to the outside interface of the FWSM.

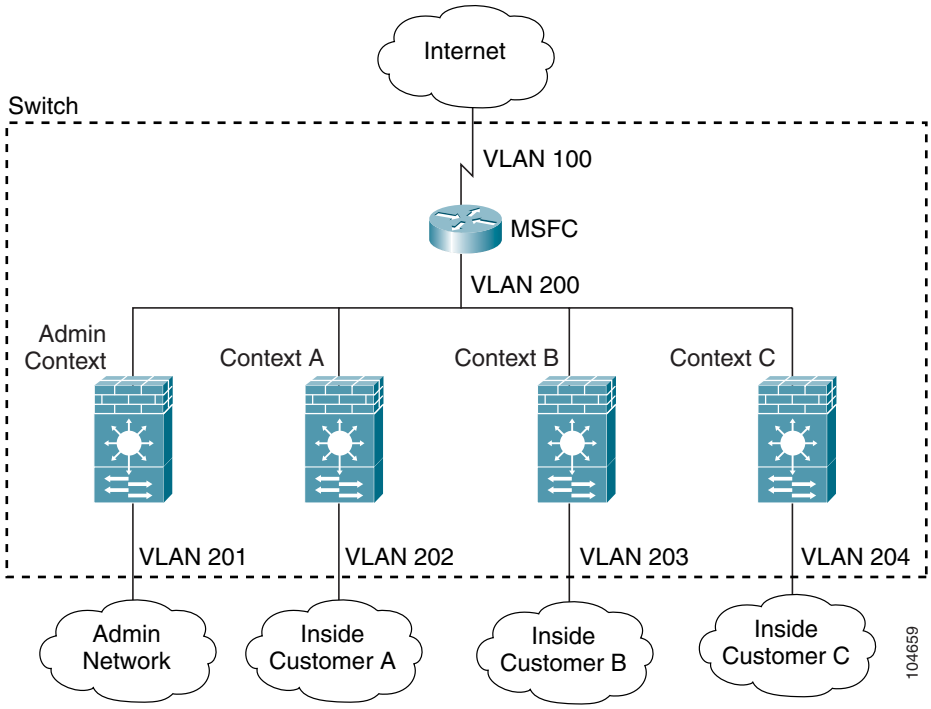
In the left-hand example, the MSFC routes between VLANs 201, 301, 302, and 303, and no inside traffic goes through the FWSM unless it is destined for the Internet. In the right-hand example, the FWSM processes and protects all traffic between the inside VLANs 201, 202, and 203.

Figure 1-2 MSFC Placement



For multiple context mode, if you place the MSFC behind the FWSM, you should only connect it to a single context. If you connect the MSFC to multiple contexts, the MSFC will route between the contexts, which might not be your intention. The typical scenario for multiple contexts is to use the MSFC in front of all the contexts to route between the Internet and the switched networks (see Figure 1-3).

Figure 1-3 MSFC Placement with Multiple Contexts



Routed Firewall and Transparent Firewall Modes

The FWSM can run in two firewall modes:

- Routed
- Transparent

In routed mode, the FWSM is considered to be a router hop in the network. It performs NAT between connected networks, and can use OSPF or passive RIP (in single context mode). Routed mode supports up to 256 interfaces per context or in single mode, with a maximum of 1000 interfaces divided between all contexts.

In transparent mode, the FWSM acts like a “bump in the wire,” or a “stealth firewall,” and is not a router hop. The FWSM connects the same network on its inside and outside interfaces, but each interface must be on a different VLAN. No dynamic routing protocols or NAT are required. However, like routed mode, transparent mode also requires ACLs to allow traffic through. Transparent mode can also optionally use EtherType ACLs to allow non-IP traffic. Transparent mode only supports two interfaces, an inside interface and an outside interface.

You might use a transparent firewall to simplify your network configuration. Transparent mode is also useful if you want the firewall to be invisible to attackers. You can also use a transparent firewall for traffic that would otherwise be blocked in routed mode. For example, a transparent firewall can allow multicast streams using an EtherType ACL.

See Chapter 7, "Configuring Bridging Parameters and ARP Inspection," for more information.

Security Contexts

You can partition a single FWSM into multiple virtual firewalls, known as security contexts. Each context is an independent system, with its own security policy, interfaces, and administrators. Multiple contexts are similar to having multiple stand-alone firewalls.

Each context has its own configuration that identifies the security policy, interfaces, and almost all the options you can configure on a stand-alone firewall. If desired, you can allow individual context administrators to implement the security policy on the context. Some resources are controlled by the overall system administrator, such as VLANs and system resources, so that one context cannot affect other contexts inadvertently.

The system administrator adds and manages contexts by configuring them in the system configuration, which identifies basic settings for the module. The system administrator has privileges to manage all contexts. The system configuration does not include any network interfaces or network settings for itself; rather, when the system needs to access network resources (such as downloading the contexts from the server), it uses one of the contexts that is designated as the admin context.

The admin context is just like any other context, except that when a user logs into the admin context (for example, over an SSH connection), then that user has system administrator rights, and can access the system configuration and all other context configurations. Typically, the admin context provides network access to network-wide resources, such as a syslog server or context configuration server.

With the default software license, you can run up to two security contexts plus the admin context. For more contexts, you must purchase a license.

Note You can run all your contexts in routed mode or transparent mode; you cannot run some contexts in one mode and others in another.

Note Multiple context mode supports static routing only.

See Chapter 5, "Managing Security Contexts," for more information.

Configuring the Switch for the Firewall Services Module

This chapter describes how to configure the Catalyst 6500 series switch or the Cisco 7600 series router for use with the Firewall Services Module (FWSM). Before completing the procedures in this chapter, configure the basic properties of your switch, including assigning VLANs to interfaces, according to the documentation that came with your switch.

This chapter includes the following sections:

Switch Overview

You can install the FWSM in the Catalyst 6500 series switches or the Cisco 7600 series routers. The configuration of both series is identical, and the series are referred to generically in this guide as the "switch." The switch includes a switch (the supervisor engine) as well as a router (the Multilayer Switch Feature Card (MSFC)).

The switch supports two software modes:

- Cisco IOS software on both the switch supervisor and the integrated MSFC router (known as supervisor IOS).
- Catalyst Operating System (OS) on the supervisor, and Cisco IOS software on the MSFC.

Both modes are described in this guide.

See the "Using the MSFC" section on page 1-9 for more information about the MSFC.

Note For each FWSM in a switch using Cisco IOS software, the SPAN reflector feature is enabled. This feature enables multicast traffic (and other traffic that requires central rewrite engine) to be switched when coming from the FWSM. The SPAN reflector feature uses one SPAN session. To disable this feature, enter the following command:

```
Router(config)# no monitor session servicemodule
```

Verifying the Module Installation

To verify that the switch acknowledges the FWSM and has brought it online, view the module information according to your operating system:

- Cisco IOS Software

```
Router> show module [mod-num | all]
```

The following example shows the output of the **show module** command:

```
Router> show module
Mod Ports Card Type                                Model              Serial No.
--- ----- ---------------------------------------- ------------------ -----------
  1    2  Catalyst 6000 supervisor 2 (Active)    WS-X6K-SUP2-2GE    SAD0444099Y
  2   48  48 port 10/100 mb RJ-45 ethernet       WS-X6248-RJ-45     SAD03475619
  3    2  Intrusion Detection System             WS-X6381-IDS       SAD04250KV5
  4    6  Firewall Module                        WS-SVC-FWM-1       SAD062302U4
```

- Catalyst OS

```
Console> show module [mod-num]
```

The following example shows the output of the **show module** command:

```
Console> show module
Mod Slot Ports Module-Type               Model               Sub Status
--- ---- ----- ------------------------- ------------------- --- ------
1   1    2     1000BaseX Supervisor      WS-X6K-SUP1A-2GE    yes ok
15  1    1     Multilayer Switch Feature WS-F6K-MSFC         no  ok
4   4    2     Intrusion Detection Syste WS-X6381-IDS        no  ok
5   5    6     Firewall Module           WS-SVC-FWM-1        no  ok
6   6    8     1000BaseX Ethernet        WS-X6408-GBIC       no  ok
```

Note The **show module** command shows six ports for the FWSM; these are internal ports that are grouped together as an EtherChannel. See the "Customizing the FWSM Internal Interface" section on page 2-11 for more information.

Assigning VLANs to the Firewall Services Module

This section describes how to assign VLANs to the FWSM. The FWSM does not include any external physical interfaces. Instead, it uses VLAN interfaces. Assigning VLANs to the FWSM is similar to assigning a VLAN to a switch port; the FWSM includes an internal interface to the Switch Fabric Module (if present) or the shared bus.

See the following topics:

Prerequisites

Follow these steps to make sure you can use the VLANs on the FWSM. See the documentation for the switch for detailed information.

1. Add the VLANs to the switch.

   If you do not add the VLANs to the switch before you assign them to the FWSM, the VLANs are stored in the supervisor engine database and are sent to the FWSM as soon as they are added to the switch.

   The VLANs cannot be reserved VLANs.

   - **Cisco IOS Software**

     To add the VLAN, enter the **vlan** *vlan_number* command.

   - **Catalyst OS**

     To add the VLAN, enter the **set vlan** *vlan_number* command.

2. Assign the VLANs to switch ports.

   - **Cisco IOS Software**

     To assign a VLAN to a port, enter:

     ```
     router(config)# interface type slot/port
     router(config-if)# switchport
     router(config-if)# switchport mode access
     router(config-if)# switchport access vlan vlan_id
     ```

   - **Catalyst OS**

     To assign a VLAN to a port, enter the **set vlan** *vlan_number mod/ports* command. This command both creates the VLAN (if you have not already done so) and assigns it to a port.

Note If you are using FWSM failover within the same switch chassis, do not assign the VLAN(s) you are reserving for failover and stateful communications to a switch port. However, if you are using failover between chassis, you must include the VLANs in the trunk port between the chassis.

3. Assign VLANs to the FWSM before you assign them to the MSFC.

   VLANs that do not satisfy this condition are discarded from the range of VLANs that you attempt to assign on the FWSM. See the "Adding Switched Virtual Interfaces to the MSFC" section on page 2-5 for more information.

Assigning VLANs in Cisco IOS Software

In Cisco IOS software, create one or more firewall VLAN groups, and then assign the groups to the FWSM. For example, you can assign all the VLANs to one group, or you can create an inside group and an outside group, or you can create a group for each customer.

You cannot assign the same VLAN to multiple firewall groups; however, you can assign multiple firewall groups to an FWSM. VLANs that you want to assign to multiple FWSMs, for example, can reside in a separate group from VLANs that are unique to each FWSM.

To assign VLANs to the FWSM, follow these steps:

Step 1 To assign VLANs to a firewall group, enter the following command:

```
Router(config)# firewall vlan-group firewall_group vlan_range
```

The *vlan_range* can be one or more VLANs (1 to 1000 and from 1025 to 4094) identified in one of the following ways:

- A single number (*n*)
- A range (*n-x*)

Separate numbers or ranges by commas. For example, enter the following numbers:

```
5,7-10,13,45-100
```

Note Routed ports and WAN ports consume internal VLANs, so it is possible that VLANs in the 1020-1100 range might already be in use.

Step 2 To assign the firewall groups to the FWSM, enter the following command:

```
Router(config)# firewall module module_number vlan-group firewall_group
```

The *firewall_group* is one or more group numbers:

- A single number (*n*)
- A range (*n-x*)

Separate numbers or ranges by commas. For example, enter the following numbers:

```
5,7-10
```

This example shows how you can create three firewall VLAN groups: one for each FWSM, and one that includes VLANs assigned to both FWSMs. See the "Prerequisites" section on page 2-3 for more information about adding VLANs to the switch.

```
Router(config)# vlan 55-57,70-85,100
Router(config-vlan)# exit
Router(config)# firewall vlan-group 50 55-57
Router(config)# firewall vlan-group 51 70-85
Router(config)# firewall vlan-group 52 100
Router(config)# firewall module 5 vlan-group 50,52
Router(config)# firewall module 8 vlan-group 51,52
```

To view the group configuration, enter the following command:

```
Router# show firewall vlan-group
Group vlans
----- ------
   50 55-57
   51 70-85
   52 100
```

To view VLAN group numbers for all modules, enter the following command:

```
Router# show firewall module
Module Vlan-groups
  5    50,52
  8    51,52
```

Assigning VLANs in Catalyst OS Software

In Catalyst OS software, you assign a list of VLANs to the FWSM. You can assign the same VLAN to multiple FWSMs if desired.

To assign VLANs to the FWSM, enter the following command:

```
Console> (enable) set vlan vlan_list firewall-vlan mod_num
```

The *vlan_list* can be one or more VLANs (1 to 1000 and from 1025 to 4094) identified in one of the following ways:

- A single number (*n*)
- A range (*n-x*)

Separate numbers or ranges by commas. For example:

```
5,7-10,13,45-100
```

Note Routed ports and WAN ports consume internal VLANs, so it is possible that VLANs in the 1020-1100 range might already be in use.

This example shows a typical configuration:

```
Console> (enable) set vlan 55-57
Console> (enable) set vlan 70-85
Console> (enable) set vlan 100
Console> (enable) set vlan 55-57,100 firewall-vlan 5
Console> (enable) set vlan 70-85,100 firewall-vlan 8
```

To view the VLANs assigned to the FWSM, enter the following command:

```
Console> show vlan firewall-vlan 5
Secured vlans by firewall module 5
55-57, 100
```

Adding Switched Virtual Interfaces to the MSFC

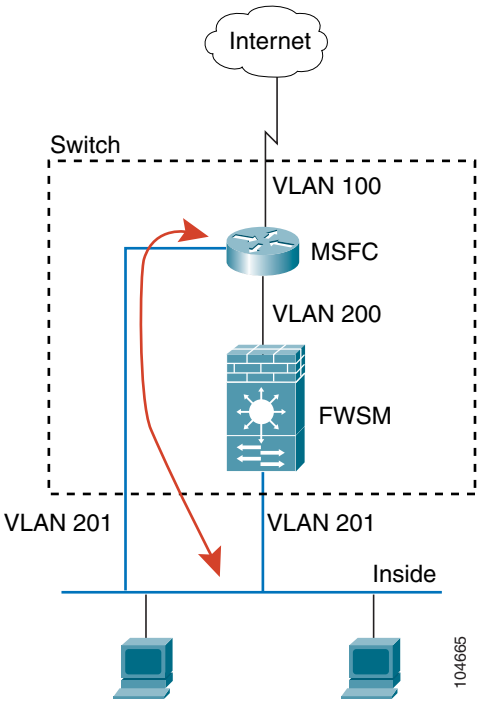
A VLAN defined on the MSFC is called a switched virtual interface (SVI). If you assign the VLAN used for the SVI to the FWSM (see the "Assigning VLANs to the Firewall Services Module" section on page 2-2), then the MSFC routes between the FWSM and other Layer 3 VLANs.

This section includes the following topics:

SVI Overview

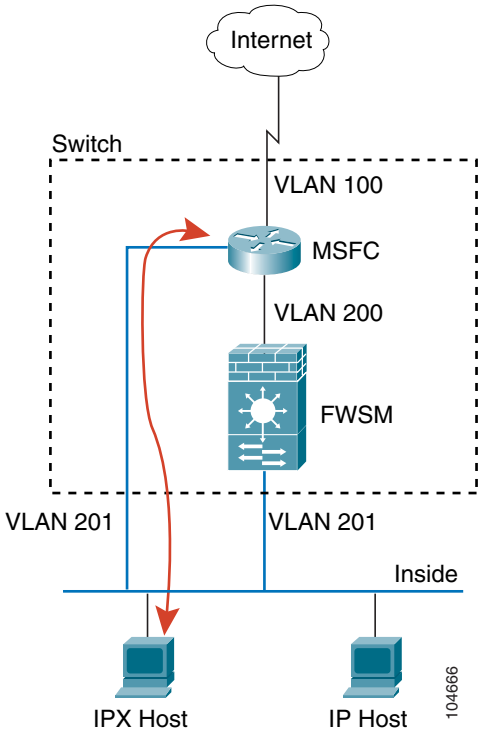
For security reasons, by default, only one SVI can exist between the MSFC and the FWSM. For example, if you misconfigure the system with multiple SVIs, you could accidentally allow traffic to pass around the FWSM by assigning both the inside and outside VLANs to the MSFC (see Figure 2-1).

Figure 2-1 Multiple SVI Misconfiguration



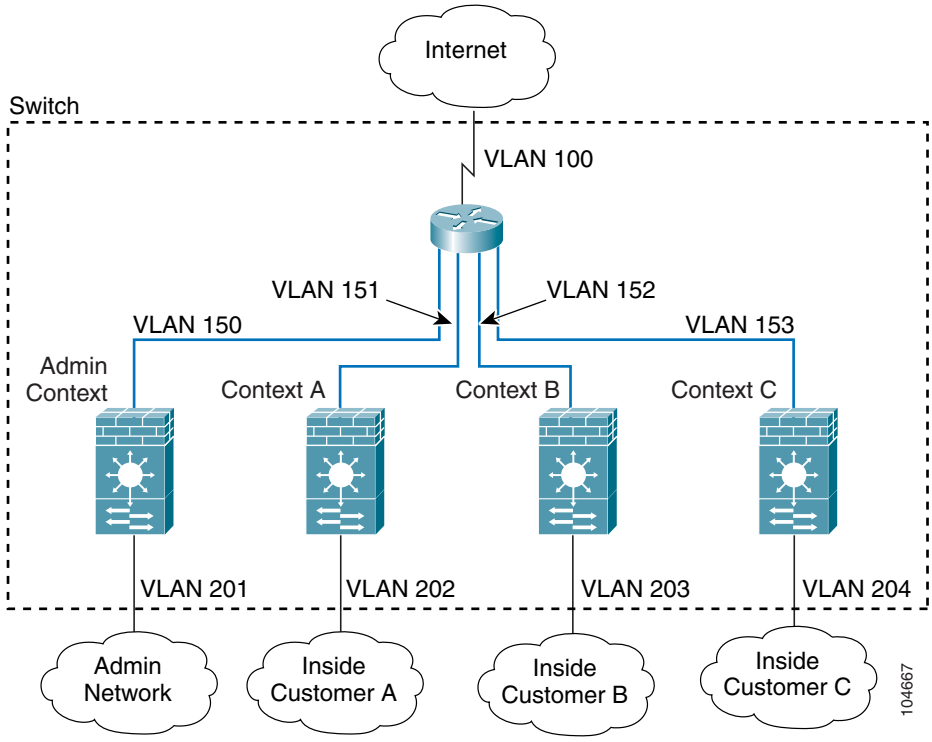
However, you might need to bypass the FWSM in some network scenarios. Figure 2-2 shows an IPX host on the same Ethernet segment as IP hosts. Because the FWSM in routed firewall mode only handles IP traffic and drops other protocol traffic like IPX (transparent firewall mode can optionally allow non-IP traffic), you might want to bypass the FWSM for IPX traffic. Make sure to configure the MSFC with an ACL that allows only IPX traffic to pass on VLAN 201.

Figure 2-2 Multiple SVIs for IPX



For transparent firewalls in multiple context mode, you need to use multiple SVIs because each context requires a unique VLAN on its outside interface (see Figure 2-3). You might also choose to use multiple SVIs in routed mode so you do not have to share a single VLAN for the outside interface.

Figure 2-3 Multiple SVIs in Multiple Context Mode



Configuring SVIs for Cisco IOS Software on the Supervisor

If you are running Cisco IOS software on the supervisor, follow these steps to add an SVI to the MSFC:

Step 1 (Optional) To allow you to add more than one SVI to the FWSM, enter the following command:

```
Router(config)# firewall multiple-vlan-interfaces
```

Step 2 To add a VLAN interface to the MSFC, enter the following command:

```
Router(config)# interface vlan vlan_number
```

Step 3 To set the IP address for this interface on the MSFC, enter the following command:

```
Router(config-if)# ip address address mask
```

Step 4 To enable the interface, enter the following command:

```
Router(config-if)# no shut
```

This example shows a typical configuration with multiple SVIs:

```
Router(config)# vlan 55-57,70-85
Router(config-vlan)# exit
Router(config)# firewall vlan-group 50 55-57
Router(config)# firewall vlan-group 51 70-85
Router(config)# firewall module 8 vlan-group 50-51
Router(config)# firewall multiple-vlan-interfaces
Router(config)# interface vlan 55
Router(config-if)# ip address 10.1.1.1 255.255.255.0
Router(config-if)# no shut
Router(config-if)# interface vlan 56
Router(config-if)# ip address 10.1.2.1 255.255.255.0
Router(config-if)# no shut
Router(config-if)# end
Router#
```

To view your SVI configuration, enter the following command:

```
Router# show int vlan 55
Vlan55 is up, line protocol is up
  Hardware is EtherSVI, address is 0008.20de.45ca (bia 0008.20de.45ca)
  Internet address is 55.1.1.1/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  ARP type:ARPA, ARP Timeout 04:00:00
  Last input never, output 00:00:08, output hang never
  Last clearing of "show interface" counters never
  Input queue:0/75/0/0 (size/max/drops/flushes); Total output drops:0
  Queueing strategy:fifo
  Output queue :0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  L2 Switched:ucast:196 pkt, 13328 bytes - mcast:4 pkt, 256 bytes
  L3 in Switched:ucast:0 pkt, 0 bytes - mcast:0 pkt, 0 bytes mcast
  L3 out Switched:ucast:0 pkt, 0 bytes
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     4 packets output, 256 bytes, 0 underruns
     0 output errors, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Configuring SVIs for Catalyst OS on the Supervisor

If you are running Catalyst OS on the supervisor, follow these steps to add an SVI to the MSFC:

Step 1 (Optional) To allow you to add more than one SVI to the FWSM. enter the following command:

```
Console> (enable) set firewall multiple-vlan-interfaces enable
```

To disable this setting, enter the following command:

```
Console> (enable) set firewall multiple-vlan-interfaces disable
```

Step 2 To access the MSFC interface, enter one of the following commands:

```
Console> (enable) switch console
```

or

```
Console> (enable) session {15 | 16}
```

If you are accessing the switch using Telnet or SSH, you must use the **session** command.

Step 3 To enter enable mode and then configuration mode on the MSFC, enter the following commands:

```
Router> enable
Router# configure terminal
```

Step 4 To add a VLAN interface to the MSFC, enter the following command:

```
Router(config)# interface vlan vlan_number
```

Step 5 To set the IP address for this interface on the MSFC, enter the following command:

```
Router(config-if)# ip address address mask
```

Step 6 To enable the interface, enter the following command:

```
Router(config-if)# no shut
```

Step 7 To return to privileged EXEC mode, enter the following command:

```
Router(config-if)# end
```

Step 8 To return to the switch CLI, enter **Ctrl-C** three times.

This example shows a typical configuration:

```
Console> (enable) set vlan 55-57
Console> (enable) set vlan 70-85
Console> (enable) set vlan 55-57,70-85 firewall-vlan 8
Console> (enable) set firewall multiple-vlan-interfaces enable
Console> (enable) switch console
Router> enable
Password: ******
Router# configure terminal
Router(config)# interface vlan 55
Router(config-if)# ip address 10.1.1.1 255.255.255.0
Router(config-if)# no shut
Router(config-if)# interface vlan 56
Router(config-if)# ip address 10.1.2.1 255.255.255.0
Router(config-if)# no shut
Router(config-if)# end
Router# ^C^C^C
Console> (enable)
```

To view your SVI configuration, enter the following command at the MSFC prompt:

```
Router# show int vlan 55
Vlan55 is up, line protocol is up
  Hardware is EtherSVI, address is 0008.20de.45ca (bia 0008.20de.45ca)
  Internet address is 55.1.1.1/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  ARP type:ARPA, ARP Timeout 04:00:00
  Last input never, output 00:00:08, output hang never
```

```
Last clearing of "show interface" counters never
Input queue:0/75/0/0 (size/max/drops/flushes); Total output drops:0
Queueing strategy:fifo
Output queue :0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
L2 Switched:ucast:196 pkt, 13328 bytes - mcast:4 pkt, 256 bytes
L3 in Switched:ucast:0 pkt, 0 bytes - mcast:0 pkt, 0 bytes mcast
L3 out Switched:ucast:0 pkt, 0 bytes
   0 packets input, 0 bytes, 0 no buffer
   Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
   4 packets output, 256 bytes, 0 underruns
   0 output errors, 0 interface resets
   0 output buffer failures, 0 output buffers swapped out
```

Customizing the FWSM Internal Interface

The connection between the FWSM and the switch is a 6-GB 802.1Q trunking EtherChannel. This EtherChannel is automatically created when you install the FWSM. On the FWSM side, two network processors (NPs) connect to three Gigabit Ethernet interfaces each, and these interfaces comprise the EtherChannel. The switch distributes traffic to the interfaces in the EtherChannel according to a distribution algorithm based on session information; load sharing is not performed on a per-packet basis, but rather on a flow basis. In some cases, the algorithm assigns traffic unevenly between the interfaces and, therefore, between the two NPs. Aside from not utilizing the full processing potential of the FWSM, consistent inequity can result in unexpected behavior when you apply resource management to multiple contexts (see the "Configuring a Class" section on page 5-14 for more information.) To make changes to the algorithm see the command for your operating system:

- Cisco IOS Software

  ```
  Router(config)# port-channel load-balance {dst-ip | dst-mac | dst-port | src-dst-ip |
  src-dst-mac | src-dst-port | src-ip | src-mac | src-port}
  ```

  The default is **src-dst-ip**.

- Catalyst OS

  ```
  Console> (enable) set port channel all distribution {ip | mac | session |
  ip-vlan-session} [source | destination | both]
  ```

  The default is **ip both**.

Configuring the Switch for Failover

To configure the switch for failover, see the following topics:

- Assigning VLANs to the Secondary Firewall Services Module, page 2-12
- Adding a Trunk Between a Primary Switch and Secondary Switch, page 2-12
- Ensuring Compatibility with Transparent Firewall Mode, page 2-12

Assigning VLANs to the Secondary Firewall Services Module

Because both units require the same access to the inside and outside networks, you must assign the same VLANs to both FWSMs on the switch(es). See the "Assigning VLANs to the Firewall Services Module" section on page 2-2.

Adding a Trunk Between a Primary Switch and Secondary Switch

If you are using inter-switch failover (see the "Module Placement" section on page 15-4), then you need to configure an 802.1Q VLAN trunk between the two switches. The trunk should have the following characteristics:

- The trunk must carry all firewall VLANs, including the failover and state VLANs.
- Because this trunk also accommodates FWSM traffic when a module fails, this trunk should be at least as large as the maximum amount of traffic you expect to be inspected by the FWSM. The FWSM has an internal 6-Gbps EtherChannel to the switch, so if the FWSM runs at full capacity, the trunk between the two devices needs to include at least six 1-Gbps interfaces. EtherChannel aggregates the bandwidth of up to eight compatibly configured ports into a single logical link. If you do not have the ports to spare, you can create a smaller trunk; however, you might experience decreased performance.
- The trunk should have QoS enabled so that failover VLAN packets, which have the COS value of 5 (higher priority), are treated with higher priority in these ports.

To configure the EtherChannel and trunk, see the documentation for your switch.

Ensuring Compatibility with Transparent Firewall Mode

To avoid loops when you use failover in transparent mode, use switch software that supports BPDU forwarding. Catalyst OS Version 8.2(1) and Cisco IOS Version 12.2(17)SXA allow BPDUs automatically.

Managing the Firewall Services Module Boot Partitions

This section describes how to reset the FWSM from the switch, and how to manage the boot partitions on the Compact Flash card. This section includes the following topics:

Flash Memory Overview

The FWSM has a 128-MB Compact Flash card ("Flash memory") that stores the operating system, configurations, and other data. The Flash memory includes six partitions, called **cf:***n* in Cisco IOS and Catalyst operating system commands:

- Maintenance partition (**cf:1**)—Contains the maintenance image. Use the maintenance partition to upgrade or install application images if you cannot boot into the application partition, to reset the application image password, or to display the crash dump information.
- Network configuration partition (**cf:2**)—Contains the network configuration of the maintenance image. The maintenance partition requires IP settings so that the FWSM can reach the TFTP server to download software images.
- Crash dump partition (**cf:3**)—Stores the crash dump information.
- Application partitions (**cf:4** and **cf:5**)—Stores the software image, system configuration, and PDM for FWSM. By default, Cisco installs the images on **cf:4**. You can use **cf:5** as a test partition. For example, if you want to upgrade your software, you can install the new software on **cf:5**, but maintain the old software as a backup in case you have problems.
- Security context partition (**cf:6**)—64 MB are dedicated to this partition, which stores security context configurations (if desired) and RSA keys in a navigable file system. All other partitions do not have file systems that allow you to perform common tasks such as listing files. This partition is called **disk** when using the **copy** command.

Setting the Default Boot Partition

By default, the FWSM boots from the **cf:4** application partition. However, you can choose to boot from the **cf:5** application partition or into the **cf:1** maintenance partition. To change the default boot partition, enter the command for your operating system:

- Cisco IOS Software

  ```
  Router(config)# boot device module mod_num cf:n
  ```

  Where *n* is 1 (maintenance), 4 (application), or 5 (application).

- Catalyst OS

  ```
  Console> (enable) set boot device cf:n mod_num
  ```

  Where *n* is 1 (maintenance), 4 (application), or 5 (application).

Resetting the FWSM or Booting from a Specific Partition

This section describes how to reset the FWSM or boot from a specific partition. You might need to reset the FWSM if you cannot reach it through the CLI or an external Telnet session. You might need to boot from a non-default boot partition if you need to access the maintenance partition or if you want to boot from a different software image in the backup application partition. The maintenance partition is valuable for troubleshooting.

The reset process might take several minutes.

For Cisco IOS software, when you reset the FWSM, you can also choose to run a full memory test. When the FWSM initially boots, it only runs a partial memory test. A full memory test takes approximately 6 minutes.

To reset the FWSM, see the section for your operating system:

- Resetting the FWSM in Cisco IOS Software, page 2-14
- Resetting the FWSM in Catalyst OS, page 2-14

Note To reload the FWSM when you are logged into the FWSM, enter **reload** or **reboot**. You cannot boot from a non-default boot partition with these commands.

Resetting the FWSM in Cisco IOS Software

To reset the FWSM from the switch CLI, enter the following command:

```
Router# hw-module module mod_num reset [cf:n] [mem-test-full]
```

The **cf:***n* argument is the partition, either 1 (maintenance), 4 (application), or 5 (application). If you do not specify the partition, the default partition is used (typically **cf:4**).

The **mem-test-full** option runs a full memory test, which takes approximately 6 minutes.

This example shows how to reset the FWSM installed in slot 9. The default boot partition is used.

```
Router# hw-mod mod 9 reset

Proceed with reload of module? [confirm] y
% reset issued for module 9

Router#
00:26:55:%SNMP-5-MODULETRAP:Module 9 [Down] Trap
00:26:55:SP:The PC in slot 8 is shutting down. Please wait ...
```

Resetting the FWSM in Catalyst OS

To reset the FWSM from the switch CLI, enter the following command:

```
Console> (enable) reset mod_num [cf:n]
```

where **cf:***n* is the partition, either 1 (maintenance), 4 (application), or 5 (application). If you do not specify the partition, the default partition is used (typically cf:4).

Connecting to the Firewall Services Module and Managing the Configuration

This chapter tells how to access the Firewall Services Module (FWSM) command-line interface (CLI) and manage the configuration, and contains the following sections:

Sessioning and Logging into the Firewall Services Module

This section describes how to connect or "session," to the FWSM from the switch command line, log in, access privileged mode, and then configuration mode so you can configure the FWSM. The FWSM does not have an external console port, you must session into the FWSM for initial configuration. Later, when you configure interfaces and IP addresses on the FWSM itself, you can access the FWSM CLI remotely through an FWSM interface. See Chapter 11, "Allowing Remote Management," for more information.

Without any additional configuration for user authentication, the login method consists of logging in as the default user:

1. The login password lets you access unprivileged mode.
2. To access configuration commands, you must enter privileged mode, which requires a second password (privileged mode is also known as enable mode).
3. From privileged mode, you can access configuration mode, which does not require a password.

Caution Management access to the FWSM causes a degradation in performance. We recommend that you avoid accessing the FWSM when high network performance is critical.

Note For multiple context mode, see the "Logging into the FWSM in Multiple Context Mode" section on page 5-9 for more information about logging into security contexts.

This section describes how to log in as the default user. For information about advanced configuration options for login access, see the sections for the following features:

- User authentication for CLI access—See the "Configuring Authentication for CLI Access" section on page 12-8. You can configure user authentication for accessing the CLI from Telnet, SSH, and HTTP (for the PDM for FWSM). You can also require authentication for the **enable** command.
- Command authorization—See the "Configuring Command Authorization" section on page 12-10 or "Configuring TACACS+ Command Authorization" section on page 12-13. Use command authorization (the authority to enter commands) in conjunction with CLI or enable authentication.

To session into the FWSM, log in, access privileged mode, and then configuration mode, follow these steps:

Step 1 Session into the FWSM using the command appropriate for your switch operating system:

- Cisco IOS Software

```
Router# session slot number processor 1
```

- Catalyst OS

```
Console> (enable) session module_number
```

For multiple context mode, when you session into the FWSM, you access the system configuration. See Chapter 5, "Managing Security Contexts," for more information.

Step 2 Log into the FWSM by entering the login password at the following prompt:

```
FWSM passwd:
```

By default, the password is **cisco**.

To change the password, see the "Changing the Passwords" section on page 6-1.

Step 3 To access privileged mode, enter the following command:

```
FWSM> enable
```

This command accesses the highest privilege level.

The following prompt appears:

```
Password:
```

Step 4 Enter the enable password at the prompt.

By default, the password is blank, and you can press the **Enter** key to continue. See the "Changing the Passwords" section on page 6-1 to change the enable password.

The prompt changes to:

```
FWSM#
```

To exit privileged mode, enter **disable**. You can also enter **exit** or **quit** to exit the current access mode (privileged mode, configuration mode, etc.).

Step 5 To access configuration mode, enter the following command:

```
FWSM# configure terminal
```

The prompt changes to the following:

```
FWSM(config)#
```

Managing the Configuration at the CLI

The FWSM loads the configuration from a text file, called the startup configuration. This file resides in the **flash** partition. When you enter a command, the change is made only to the running configuration in memory. You must manually save the running configuration to the startup configuration for your changes to remain after a reboot.

The information in this section applies to both single and multiple security contexts, except where noted. Additional information about contexts is in Chapter 5, "Managing Security Contexts."

See the "Backing Up the Configuration" section on page 16-7 for more information about managing configuration files.

This section includes the following topics:

Saving Configuration Changes

To save your running configuration to the startup configuration, enter the following command:

```
FWSM# copy running-config startup-config
```

For multiple context mode, context startup configurations can reside on external servers. In this case, the FWSM saves the configuration back to the server you identified in the context URL, except for an HTTP or HTTPS URL, which do not allow you to save the configuration to the server.

Note The **copy running-config startup-config** command is equivalent to the **write memory** command.

Viewing the Configuration

The following commands allow you to view the running and startup configurations.

- To view the running configuration, enter the following command:

  ```
  FWSM# show running-config
  ```

- To view the startup configuration, enter the following command:

  ```
  FWSM# show startup-config
  ```

Clearing and Removing Configuration Settings

To erase settings, enter one of the following commands.

- To clear all the configuration for a specified command and all its subcommands, enter the following command:

  ```
  FWSM# clear configurationcommand [subconfigurationcommand]
  ```

  This command clears all the current configuration for the specified configuration command. If you only want to clear the configuration for a specific subcommand, you can enter a value for *subconfigurationcommand*.

  For example, to clear the configuration for all **aaa** commands, enter the following command:

  ```
  FWSM# clear aaa
  ```

  To clear the configuration for only **aaa authentication** commands, enter the following command:

  ```
  FWSM# clear aaa authentication
  ```

- To disable the specific parameters or options of a command or subcommand, enter the following command:

  ```
  FWSM# no configurationcommand [subconfigurationcommand] qualifier
  ```

  In this case, you use the **no** command to remove the specific configuration identified by *qualifier*.

  For example, to remove a specific **nat** command, enter enough of the command to identify it uniquely as follows:

  ```
  FWSM# no nat (inside) 1
  ```

- To erase the startup configuration, enter the following command:

  ```
  FWSM# write erase
  ```

- To erase the running configuration, enter the following command:

  ```
  FWSM# clear configure all
  ```

Note In multiple context mode, if you enter **clear configure all** from the system configuration, you also remove all contexts and stop them from running.

Creating Text Configuration Files Offline

This guide describes how to use the CLI to configure the FWSM; when you save commands, the changes are written to a text file. Instead of using the CLI, however, you can edit a text file directly and paste a configuration at the configuration mode command-line prompt in its entirety, or line by line. Alternatively, you can download a text file to the FWSM Flash memory. See the "Downloading a Text Configuration" section on page 16-6 for information on downloading the configuration file to the FWSM.

In most cases, commands described in this guide are preceded by a CLI prompt. The prompt in the following example is "FWSM(config)#":

```
FWSM(config)# class gold
```

In the text configuration file you are not prompted to enter commands, so the prompt is omitted as follows:

```
class gold
```

See the "Text Configuration Files" section on page C-4 for more information about formatting the file.

Configuring the Firewall Mode

This chapter describes how to set the firewall mode to either routed mode or transparent mode, and includes the following sections:

Firewall Mode Overview

The FWSM can run in two firewall modes:

- Routed mode
- Transparent mode

In routed mode, the FWSM is considered to be a router hop in the network. It performs NAT between connected networks, and can use OSPF or passive RIP (in single context mode). Routed mode supports up to 256 interfaces per context or in single mode, with a maximum of 1000 interfaces divided between all contexts. Each interface is on a different subnet. You can share interfaces between contexts.

In transparent mode, the FWSM acts like a "bump in the wire," or a "stealth firewall," and is not a router hop. The FWSM connects the same network on its inside and outside interfaces, but each interface must be on a different VLAN. No dynamic routing protocols or NAT are required. However, like routed mode, transparent mode also requires ACLs to allow any traffic through aside from ARP packets. Transparent mode can allow certain types of traffic in an ACL that are blocked by routed mode, including unsupported routing protocols and multicast traffic. Transparent mode can also optionally use EtherType ACLs to allow non-IP traffic. Transparent mode only supports two interfaces, an inside interface and an outside interface.

This section includes the following topics:

Routed Mode Overview

This section includes the following topics:

- How Data Moves Through the FWSM in Routed Firewall Mode, page 4-3

IP Routing Support

The FWSM acts as a router between connected networks, and each interface requires an IP address on a different subnet. In single context mode, the routed firewall supports OSPF and RIP (in passive mode). Multiple context mode supports static routes only. We recommend using the advanced routing capabilities of the upstream and downstream routers, such as the MSFC, instead of relying on the FWSM for extensive routing needs.

Network Address Translation

NAT substitutes the local address on a packet with a global address that is routable on the destination network. In routed mode, you typically configure NAT for inside hosts that access an outside network, but you can optionally bypass NAT if you are using routable addresses.

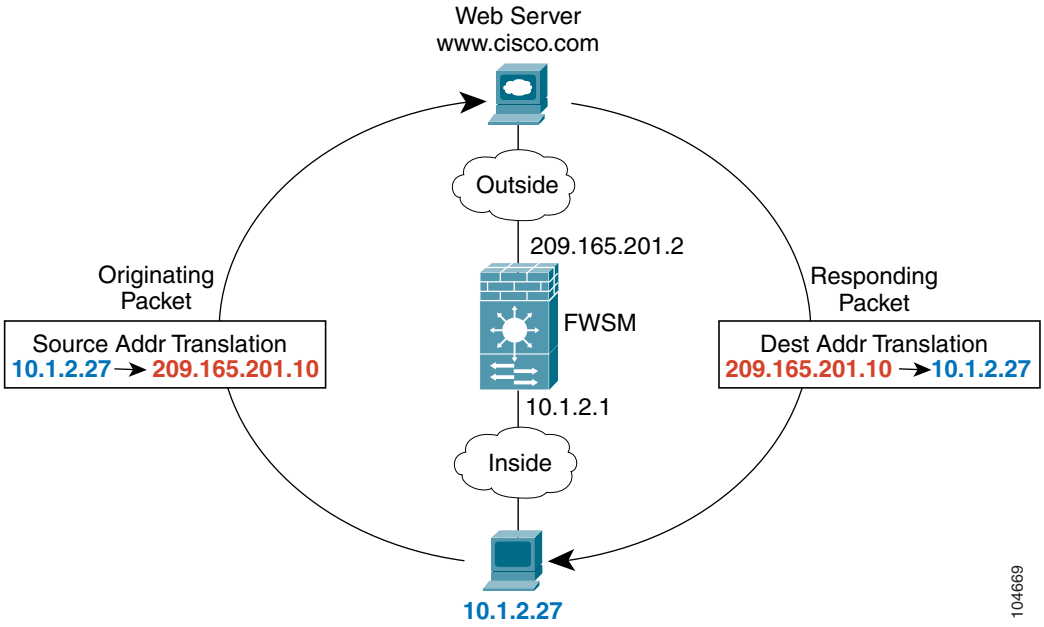
Some of the benefits of NAT include the following:

- You can use private addresses on your inside networks. Private addresses are not routable on the Internet. See the "Private Networks" section on page D-2 for more information.
- NAT hides the local addresses from other networks, so attackers cannot learn the real address of a host.
- NAT can resolve IP routing problems by supporting overlapping IP addresses.

Figure 4-1 shows a typical NAT scenario, with a private network on the inside. When the inside user sends a packet to a web server on the Internet, the local source address of the packet is changed to a routable global address. When the web server responds, it sends the response to the global address, and the firewall receives the packet. The firewall then translates the global address to the local address before sending it on to the user.

See Chapter 9, "Configuring Network Address Translation," for more information.

Figure 4-1 NAT Example



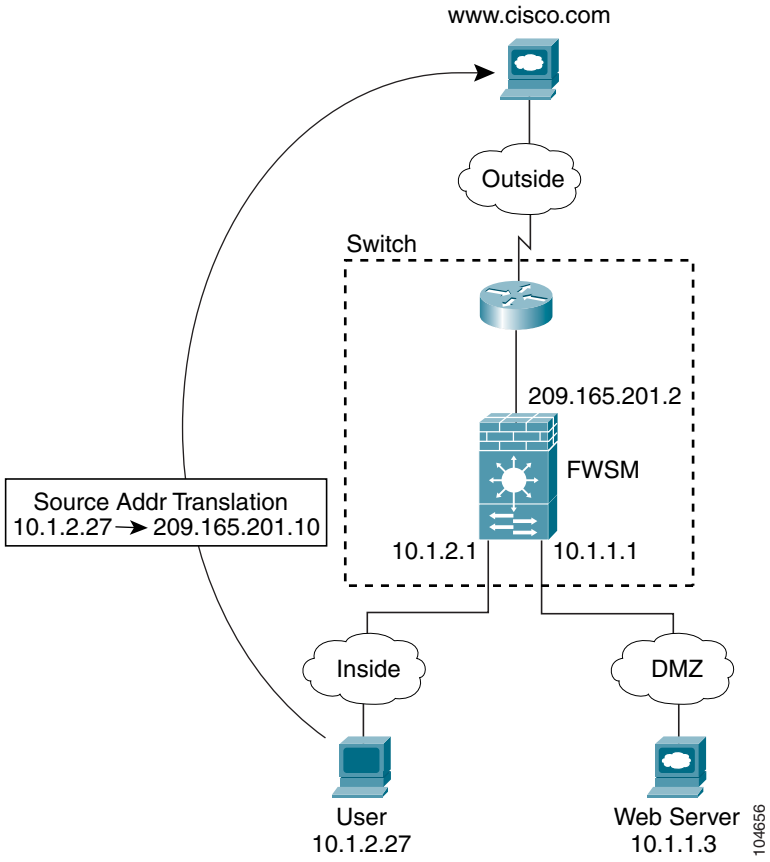
How Data Moves Through the FWSM in Routed Firewall Mode

This section describes how data moves through the FWSM in routed firewall mode, and includes the following topics:

An Inside User Visits a Website

Figure 4-2 shows an inside user accessing an outside website.

Figure 4-2 Inside to Outside



The steps below describe how data moves through the FWSM (see Figure 4-2):

1. The user on the inside network requests a web page from www.cisco.com.
2. The FWSM receives the packet, and because it is a new session, the FWSM verifies that the packet is allowed according to the terms of the security policy (ACLs, filters, AAA).

   For multiple context mode, the FWSM first classifies the packet according to either a unique VLAN or a unique destination address. In this case, the VLAN would be unique; the www.cisco.com IP address is not located uniquely within a context and is not a unique destination address.
3. The FWSM translates the local source address (10.1.2.27) to the global address 209.165.201.10, which is on the outside interface subnet.

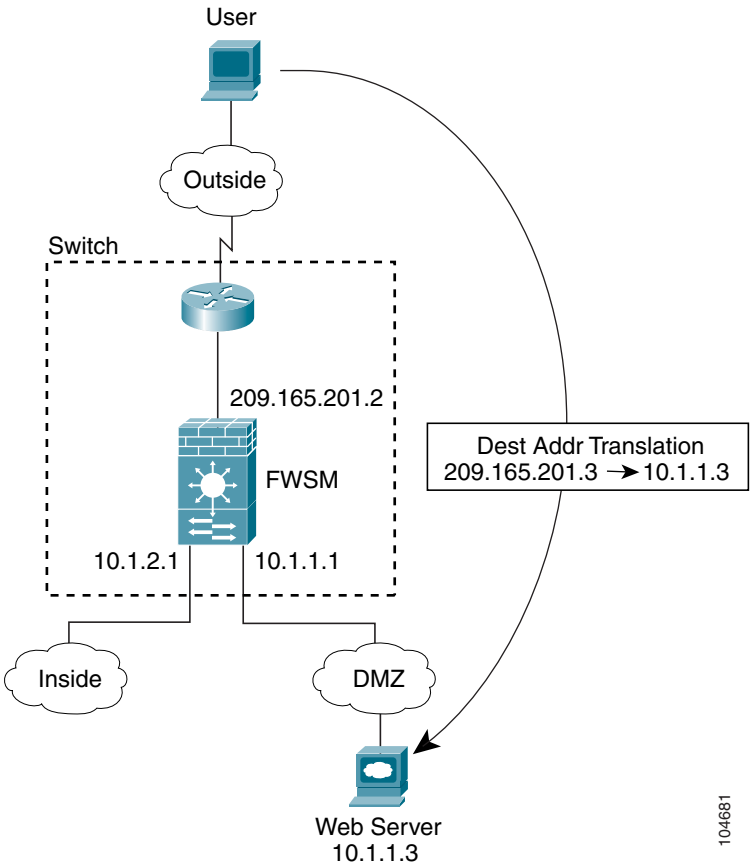
   The global address could be on any subnet, but routing is simplified when it is on the outside interface subnet.
4. The FWSM then records that a session is established and forwards the packet from the outside interface.

5. When www.cisco.com responds to the request, the packet goes through the FWSM, and because the session is already established, the packet bypasses the many lookups associated with a new connection. The fast path performs NAT by translating the global destination address to the local user address, 10.1.2.27.
6. The FWSM forwards the packet to the inside user.

An Outside User Visits a Website on the DMZ

Figure 4-3 shows an outside user accessing the DMZ website.

Figure 4-3 Outside to DMZ



The steps below describe how data moves through the FWSM (see Figure 4-3):

1. A user on the outside network requests a web page from the DMZ website using the global destination address of 209.165.201.3, which is on the outside interface subnet.
2. The FWSM receives the packet, and because it is a new session, the FWSM verifies that the packet is allowed according to the terms of the security policy (ACLs, filters, AAA).

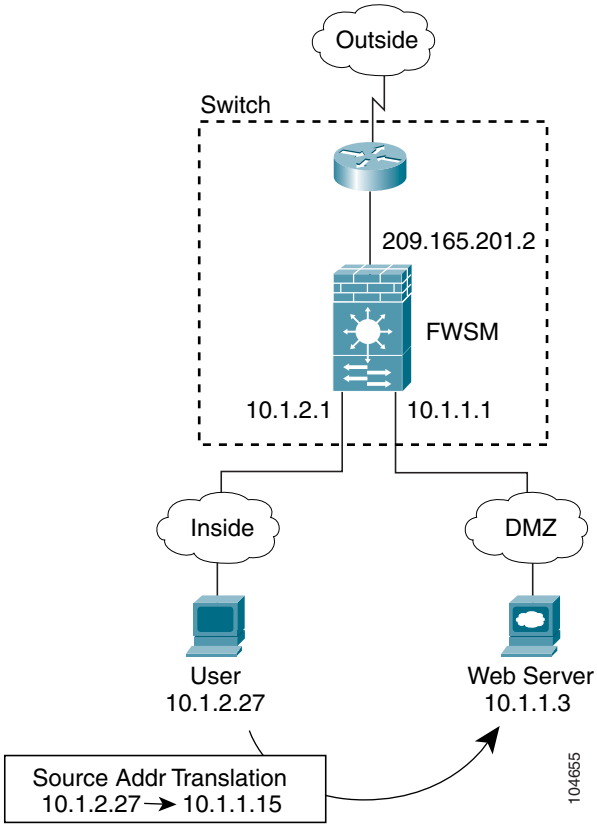
   For multiple context mode, the FWSM first classifies the packet according to either a unique VLAN or a unique destination address. In this case, even if the VLAN is not unique, the classifier "knows" that the DMZ web server address belongs to a certain context because of the NAT configuration.
3. The FWSM translates the destination address to the local address 10.1.1.3.

4. The FWSM then adds a session entry to the fast path and forwards the packet from the DMZ interface.
5. When the DMZ website responds to the request, the packet goes through the FWSM and because the session is already established, the packet bypasses the many lookups associated with a new connection. The fast path performs NAT by translating the local source address to 209.165.201.3.
6. The FWSM forwards the packet to the outside user.

An Inside User Visits a Website on the DMZ

Figure 4-4 shows an inside user accessing the DMZ website.

Figure 4-4 Inside to DMZ



The steps below describe how data moves through the FWSM (see Figure 4-4):

1. A user on the inside network requests a web page from the DMZ website using the destination address of 10.1.1.3.

   Because the DMZ is a lower security interface, the inside user can use the untranslated local address of the web server.

2. The FWSM receives the packet, and because it is a new session, the FWSM verifies that the packet is allowed according to the terms of the security policy (ACLs, filters, AAA).

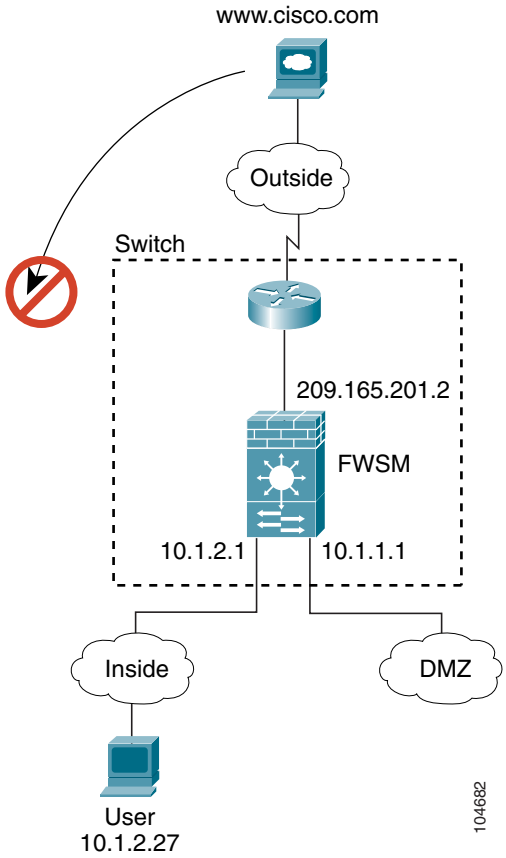
   For multiple context mode, the FWSM first classifies the packet according to either a unique VLAN or a unique destination address. In this case, the VLAN would be unique because the destination is on a different interface in the same context.

3. The FWSM translates the local source address to the global address 10.1.1.15, which is on the DMZ subnet.
4. The FWSM then records that a session is established and forwards the packet out of the DMZ interface.
5. When the DMZ web server responds to the request, the packet goes through the fast path, which allows the packet to bypass the many lookups associated with a new connection. The fast path performs NAT by translating the global destination address to the local address of the user, 10.1.2.27.
6. The FWSM forwards the packet to the inside user.

An Outside User Attempts to Access an Inside Host

Figure 4-5 shows an outside user attempting to access the inside network.

Figure 4-5 Outside to Inside



The steps below describe how data moves through the FWSM (see Figure 4-5):

1. A user on the outside network attempts to reach an inside host (assuming the host has a routable IP address).

   If the inside network uses private addresses, no outside user can reach the inside network without NAT. The outside user might attempt to reach an inside user by using an existing NAT session.

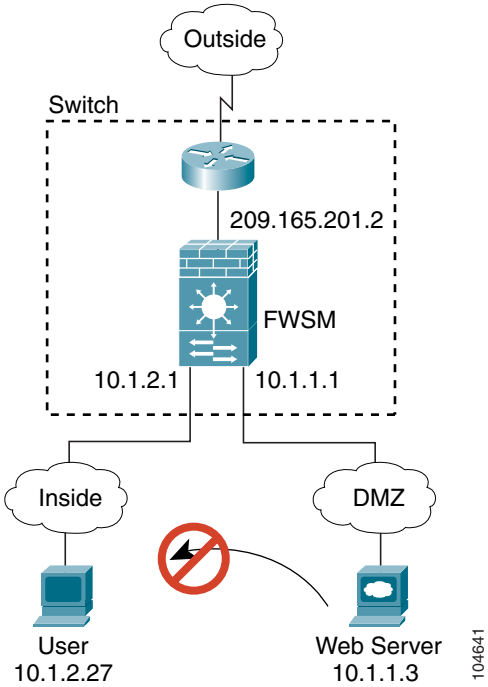
2. The FWSM receives the packet, and because it is a new session, the FWSM verifies if the packet is allowed according to the security policy (ACLs, filters, AAA).
3. The packet is denied, and the FWSM drops the packet and logs the connection attempt.

   If the outside user is attempting to attack the inside network, the FWSM employs many technologies to determine if a packet is valid for an already established session. See the "Other Protection Features" section on page 1-6 for more information.

An DMZ User Attempts to Access an Inside Host

Figure 4-6 shows a user in the DMZ attempting to access the inside network.

Figure 4-6 DMZ to Inside



The steps below describe how data moves through the FWSM (see Figure 4-6):

1. A user on the DMZ network attempts to reach an inside host. The DMZ host might know the real address of an inside host, and because the DMZ does not have to route the traffic on the internet, the private addressing scheme does not prevent routing.
2. The FWSM receives the packet, and because it is a new session, the FWSM verifies if the packet is allowed according to the security policy (ACLs, filters, AAA).
3. The packet is denied, and the FWSM drops the packet and logs the connection attempt.

Transparent Mode Overview

This section describes transparent firewall mode, and includes the following topics:

- Transparent Firewall Features, page 4-9
- Using the Transparent Firewall in Your Network, page 4-10

Transparent Firewall Features

Traditionally, a firewall is a routed hop and acts as a default gateway for hosts that connect to one of its screened subnets. A transparent firewall, on the other hand, is a Layer 2 firewall that acts like a "bump in the wire," or a "stealth firewall," and is not seen as a router hop to connected devices. The Firewall Services Module (FWSM) connects the same network on its inside and outside ports but uses different VLANs on the inside and outside.

Because the firewall is not a routed hop, you can easily introduce a transparent firewall into an existing network. You assign different VLANs to each interface, and IP readdressing is unnecessary.

Maintenance is facilitated because there are no complicated routing patterns to troubleshoot and no NAT configuration.

Even though transparent mode acts as a bridge, Layer 3 traffic, such as IP traffic, cannot pass through the FWSM unless you explicitly permit it with an extended access control list (ACL). (See the "Adding an Extended Access Control List" section on page 10-13.) The only traffic allowed through the transparent firewall without an ACL is ARP traffic. ARP traffic can be controlled by ARP inspection (see the "Configuring ARP Inspection" section on page 7-3 for more information).

In routed mode, some types of traffic cannot pass through the FWSM even if you allow it in an ACL. The transparent firewall, however, can allow any traffic through using either an extended ACL (for IP traffic) or an EtherType ACL (for non-IP traffic. See the "Adding an EtherType Access Control List" section on page 10-16 for more information).

Note The transparent mode FWSM does not pass Cisco Discovery Protocol (CDP) packets.

For example, you can allow multicast traffic such as that created by IPTV using an extended ACL. You can also establish routing protocol adjacencies through a transparent firewall; for example, you can allow OSPF, RIP, EIGRP, or BGP traffic through based on an extended ACL. Likewise, protocols like HSRP or VRRP can pass through the FWSM.

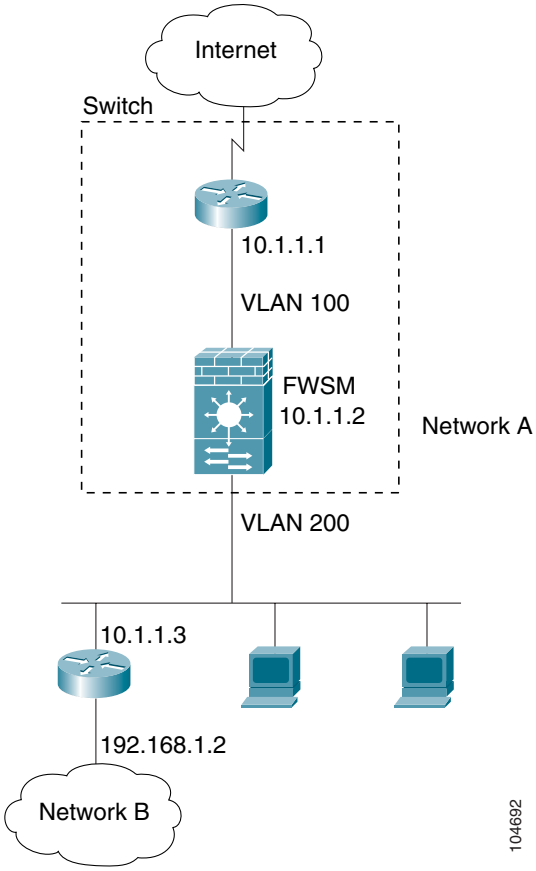
Non-IP traffic (for example IPX, BPDUs, and MPLS) can be configured to go through using an EtherType ACL.

When the FWSM runs in transparent mode, the outgoing interface of a packet is determined by performing a MAC address lookup instead of a route lookup. Route statements can still be configured, but they only apply to FWSM-originated traffic. For example, if your syslog server is located on a remote network, you must use a static route so the FWSM can reach that subnet. See the "Configuring Static Routes" section on page 8-3 for more information.

Using the Transparent Firewall in Your Network

Figure 4-7 shows a typical transparent firewall network. While the outside devices are on the same subnet as the inside devices, the VLANs are different. The inside router and hosts appear to be directly connected to the outside router. However, no traffic can bypass the FWSM because it must route between the two VLANs.

Figure 4-7 Transparent Firewall Network



Transparent Firewall Guidelines

Follow these guidelines when planning your transparent firewall network:

- The transparent FWSM uses an inside interface and an outside interface only.
- Each directly connected network must be on the same subnet.
- A management IP address is required for each context, even if you do not intend to use Telnet to the context.

  The FWSM uses this IP address as the source address for packets originating on the FWSM, such as system messages or AAA communications.

  The management IP address must be on the same subnet as the connected network.
- Do not specify the FWSM management IP address as the default gateway for connected devices; devices need to specify the router on the other side of the FWSM as the default gateway.
- Each interface must be a different VLAN interface.
- For multiple context mode, each context must use different VLANs; you cannot share a VLAN across contexts.
- For multiple context mode, each context can use the same (overlapping) subnet or different subnets. Make sure that the upstream router performs NAT if you use overlapping subnets.
- Dynamic routing protocols are neither required nor supported.

  You can, however, add static routes.
- NAT is not supported.

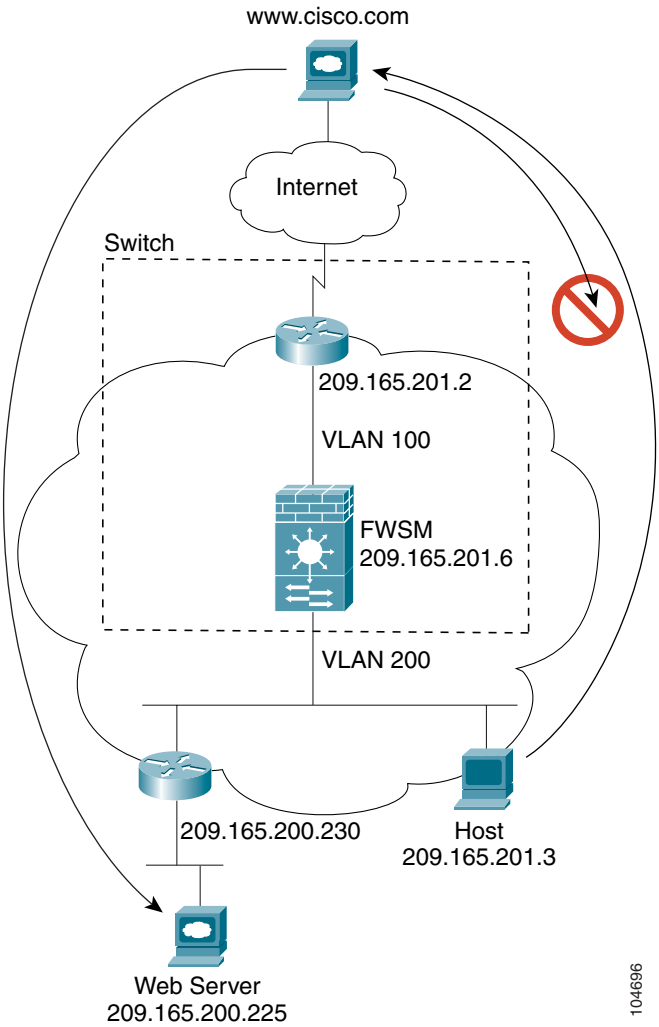
  NAT is performed on the upstream router. However, you can configure some parameters available only in the **static** command. See the "Configuring Connection Limits for Non-NAT Configurations" section on page 6-9 for more information.
- You must use an extended ACL to allow Layer 3 traffic, such as IP traffic, through the FWSM.

  You can also optionally use an EtherType ACL to allow non-IP traffic through.

How Data Moves Through the Transparent Firewall

Figure 4-8 shows a typical transparent firewall implementation with an inside network that contains a public web server. The FWSM has an ACL so that the inside users can access Internet resources. Another ACL allows the outside users to access only the web server on the inside network.

Figure 4-8 Typical Transparent Firewall Data Path



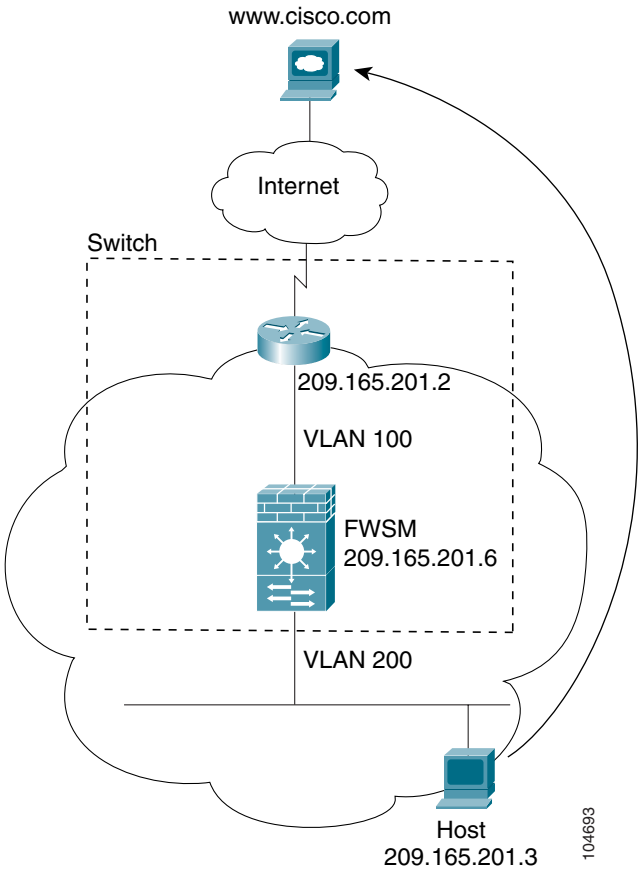
The following sections describe how data moves through the FWSM:

- An Inside User Visits a Website, page 4-13
- An Outside User Visits a Website on the Inside Network, page 4-14
- An Outside User Attempts to Access an Inside Host, page 4-15

An Inside User Visits a Website

Figure 4-2 shows an inside user accessing an outside website.

Figure 4-9 Inside to Outside



The steps below describe how data moves through the FWSM (see Figure 4-2):

1. The user on the inside network requests a web page from www.cisco.com.
2. The FWSM receives the packet on VLAN 200 and, because it is a new session, it verifies that the packet is allowed according to the terms of the security policy (ACLs, filters, AAA).

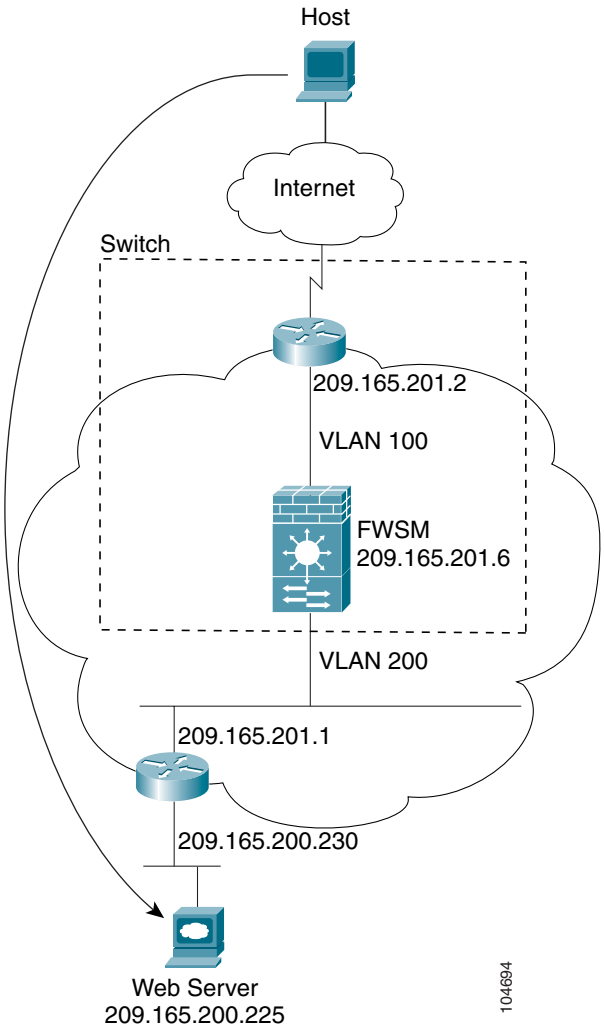
   For multiple context mode, the FWSM first classifies the packet according to either a unique VLAN or a unique destination address. In this case, the VLAN would be unique. For transparent firewall mode, each context has a unique VLAN on the inside and outside, so the IP address would not be considered.
3. The FWSM records that a session is established.
4. If the destination MAC address is in its table, the FWSM forwards the packet out of the outside interface on VLAN 100.

   If the destination MAC address is not in the FWSM table, the FWSM attempts to discover the MAC address by sending an ARP request and a ping. The first packet is dropped.
5. When the web server responds to the request, the packet goes through the FWSM, and because the session is already established, the packet bypasses the many lookups associated with a new connection.
6. The FWSM forwards the packet to the inside user.

An Outside User Visits a Website on the Inside Network

Figure 4-3 shows an outside user accessing the inside website.

Figure 4-10 Outside to Inside



The steps below describe how data moves through the FWSM (see Figure 4-3):

1. A user on the outside network requests a web page from the inside website.
2. The FWSM receives the packet on VLAN 100 and, because it is a new session, it verifies that the packet is allowed according to the terms of the security policy (ACLs, filters, AAA).

   For multiple context mode, the FWSM first classifies the packet according to either a unique VLAN or a unique destination address. In this case, the VLAN would be unique. For transparent firewall mode, each context has a unique VLAN on the inside and outside, so the IP address would not be considered.
3. The FWSM records that a session is established.

4. If the destination MAC address is in its table, the FWSM forwards the packet out of the inside interface on VLAN 200.

   If the destination MAC address is not in the FWSM table, the FWSM attempts to discover the MAC address by sending an ARP request and a ping. The first packet is dropped.

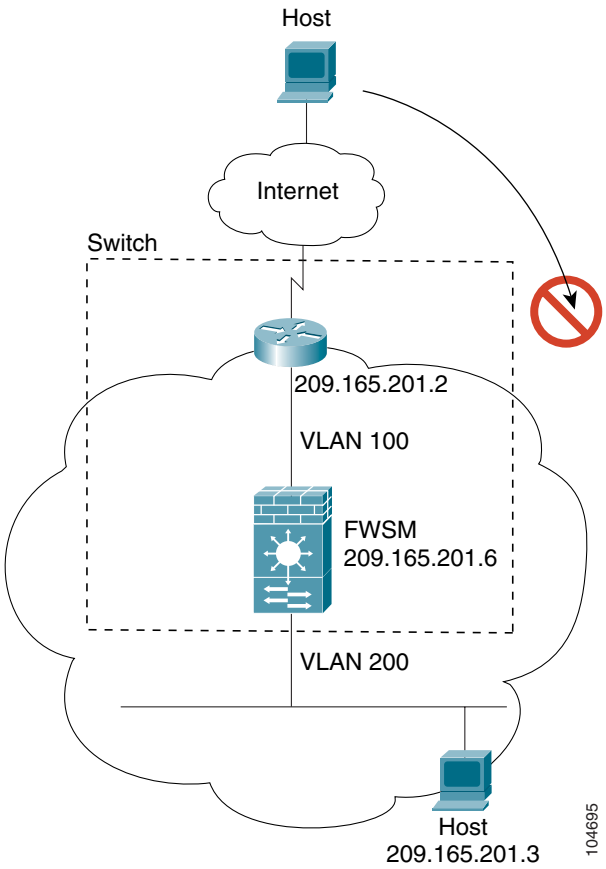
5. When the website responds to the request, the packet goes through the FWSM, and because the session is already established, the packet bypasses the many lookups associated with a new connection.

6. The FWSM forwards the packet to the outside user.

An Outside User Attempts to Access an Inside Host

Figure 4-5 shows an outside user attempting to access a host on the inside network.

Figure 4-11 Outside to Inside



The steps below describe how data moves through the FWSM (see Figure 4-5):

1. A user on the outside network attempts to reach an inside host.

2. The FWSM receives the packet and, because it is a new session, it verifies if the packet is allowed according to the terms of the security policy (ACLs, filters, AAA).

3. The packet is denied, and the FWSM drops the packet.

4. If the outside user is attempting to attack the inside network, the FWSM employs many technologies to determine if a packet is valid for an already established session. See the "Other Protection Features" section on page 1-6 for more information.

Setting the Firewall Mode

You can set the FWSM to run in routed firewall mode (the default) or transparent firewall mode.

For multiple context mode, you can use only one firewall mode for all contexts. You must set the mode in the system configuration.

When you change modes, the FWSM clears the configuration because many commands are not supported for both modes. If you already have a populated configuration, be sure to back up your configuration before changing the mode; you can use this backup for reference when creating your new configuration. See the "Backing Up the Configuration" section on page 16-7 for more information.

If you download a text configuration to the FWSM that changes the mode with the **firewall transparent** command (see below), be sure to put the command at the top of the configuration; the FWSM changes the mode as soon as it reads the command and then continues reading the configuration you downloaded. If the command is later in the configuration, the FWSM clears all the preceding lines in the configuration.

- To set the mode to transparent, enter the following command in the system execution space:

  ```
  FWSM(config)# firewall transparent
  ```

  This command also appears in each context configuration for informational purposes only; you cannot enter this command in a context.

- To set the mode to routed, enter the following command in the system execution space:

  ```
  FWSM(config)# no firewall transparent
  ```

Managing Security Contexts

This chapter tells how to configure multiple security contexts on the Firewall Services Module (FWSM), and includes the following sections:

Security Context Overview

You can partition a single FWSM into multiple virtual firewalls, known as security contexts. Each context is an independent firewall, with its own security policy, interfaces, and administrators. Multiple contexts are similar to having multiple stand-alone firewalls.

Each context has its own configuration that identifies the security policy, interfaces, and almost all the options you can configure on a stand-alone firewall. If desired, you can allow individual context administrators to implement the security policy on the context. Some resources are controlled by the overall system administrator, such as VLANs and system resources, so that one context cannot affect other contexts inadvertently.

The system administrator adds and manages contexts by configuring them in the system configuration, which identifies basic settings for the FWSM. The system administrator has privileges to manage all contexts. The system configuration does not include any network interfaces or network settings for itself; rather, when the system needs to access network resources (such as downloading the contexts from the server), it uses one of the contexts that is designated as the admin context.

The admin context is just like any other context, except that when a user logs into the admin context (for example, over an SSH connection), then that user has system administrator rights, and can access the system execution space and all other contexts. Typically, the admin context provides network access to network-wide resources, such as a syslog server or context configuration server.

This section provides an overview of security contexts, and includes the following topics:

Common Uses for Security Contexts

You might want to use multiple security contexts in the following situations:

- You are a service provider and want to sell firewall services to many customers. By enabling multiple security contexts on the FWSM, you can implement a cost-effective, space-saving solution that keeps all customer traffic separate and secure, and also eases configuration.
- You are a large enterprise or a college campus and want to keep departments completely separate.
- You are an enterprise that wants to provide distinct security policies to different departments.
- You have any network that requires more than one firewall.

Context Configuration Files

Each context has its own configuration file that identifies the security policy, interfaces, and almost all the options you can configure on a stand-alone firewall. You can store context configurations on the local **disk** partition on the Flash memory card, or you can download them from a TFTP, FTP, or HTTP(S) server.

In addition to individual security contexts, the FWSM also includes a system configuration that identifies basic settings for the FWSM, including a list of contexts. Like the single mode configuration, this configuration resides as the "startup" configuration in the **flash** partition.

The system configuration does not include any network interfaces or network settings for itself; rather, when the system needs to access network resources (such as downloading the contexts from a server), it uses one of the contexts that is designated as the admin context. The system configuration does include a specialized failover interface for failover traffic only, as well as the Ethernet Out-of-Band Channel (EOBC) to the switch, which does not require any configuration. If your system is already in multiple context mode, or if you convert from single mode, the admin context is created automatically as a file on the **disk** partition called admin.cfg. In the FWSM CLI, this context is named "admin." If you do not want to use admin.cfg as the admin context, you can change the admin context using the "Changing the Admin Context" section on page 5-20.

How the FWSM Classifies Packets

Each packet that enters the FWSM must be classified, so that the FWSM can determine to which context to send a packet. The classifier checks for the following characteristics:

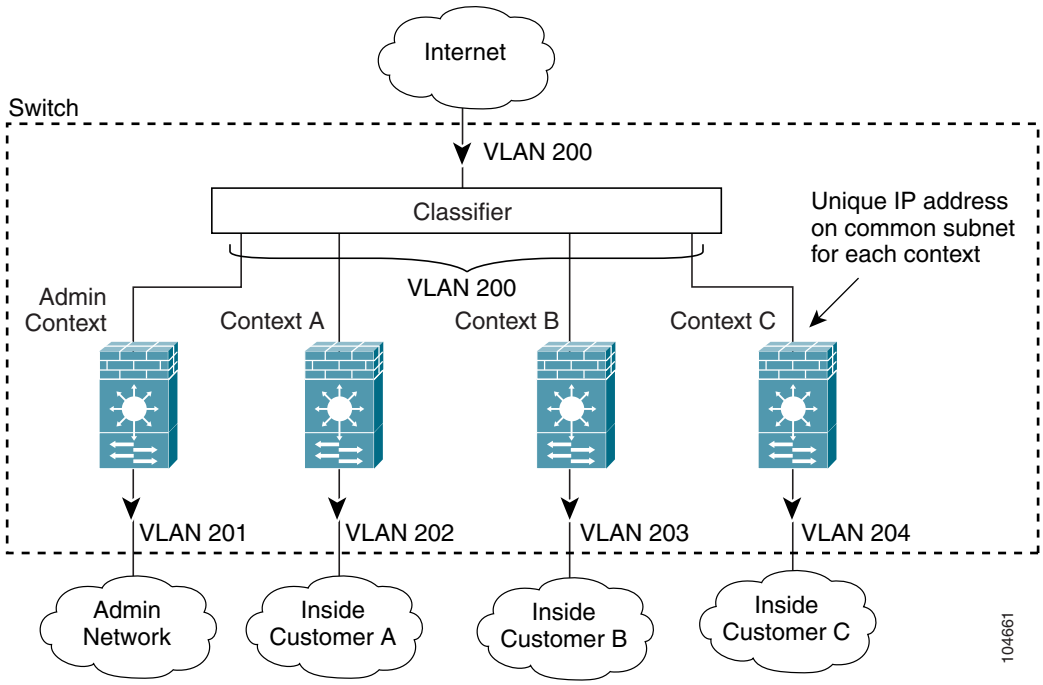
- Source interface (VLAN)
- Destination address

The FWSM uses the characteristic that is unique and not shared across contexts. For example, if you share a VLAN across contexts, then the classifier uses the IP address. See the "Sharing Resources and Interfaces Between Contexts" section on page 5-5 for more information about sharing VLANs.

The FWSM classifier only "knows" about context IP addresses that have static NAT translations or that have active NAT translations (xlates). The classifier only looks at static statements where the global interface matches the source interface of the packet.

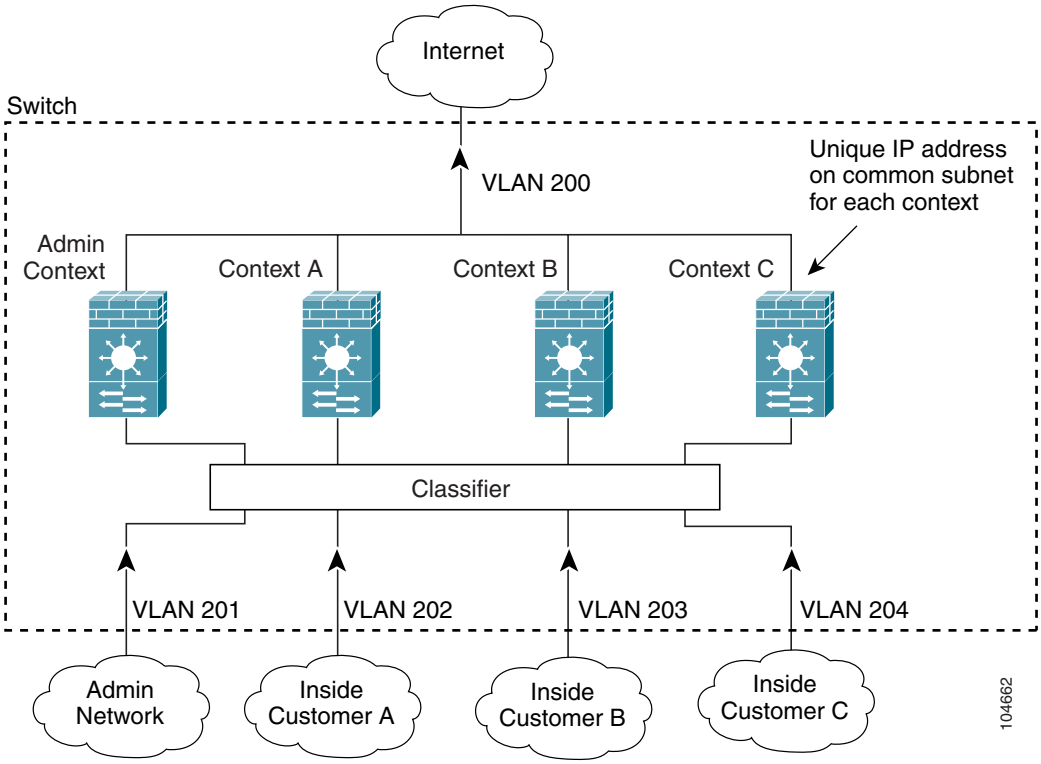
You can share a VLAN interface so long as each IP address space on that VLAN is unique, or you can have overlapping IP addresses so long as the VLANs are unique. Figure 5-1 shows multiple contexts sharing an outside VLAN, while the inside VLANs are unique, allowing overlapping IP addresses.

Figure 5-1 Multiple Security Contexts



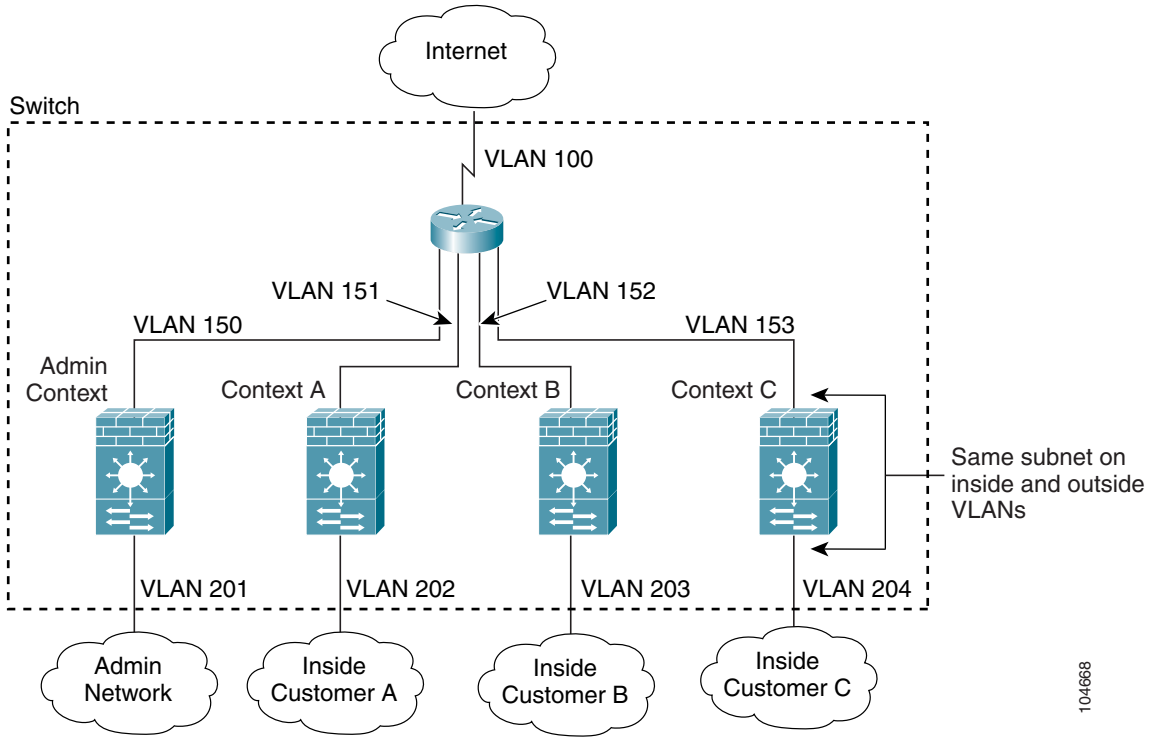
Note that all new incoming traffic must be classified, even from inside networks (see Figure 5-2).

Figure 5-2 Incoming Traffic from Inside Networks



For transparent firewalls, interfaces do not have IP addresses, so you must use unique VLANs (see Figure 5-3):

Figure 5-3 Transparent Firewall Contexts



IP Routing Support

Security contexts support only static routes. You cannot enable OSPF or RIP in multiple context mode.

Sharing Resources and Interfaces Between Contexts

The FWSM allows you to share an interface between contexts. Typically in routed mode, you share the outside interface to conserve VLANs. You can also share inside VLANs to share resources between contexts, or you can place the shared resource on a single context and provide access to that resource to other contexts.

This section includes the following topics:

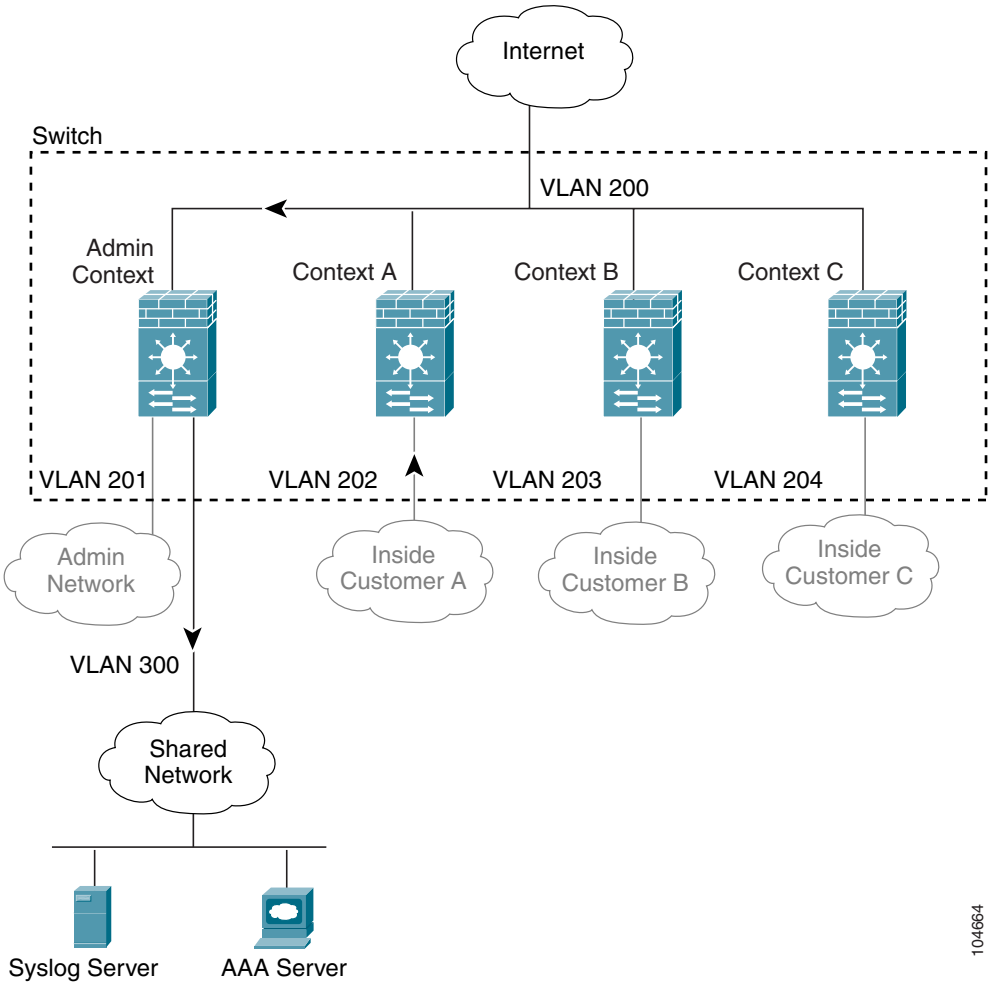
- Sharing Resources, page 5-6
- Shared Interface Limitations, page 5-7

Sharing Resources

If you have a server that needs to be accessed by multiple contexts (such as a AAA server or a syslog server), then you can choose to place the server on one context network to which all other contexts have access, or you can place the server on a shared inside VLAN.

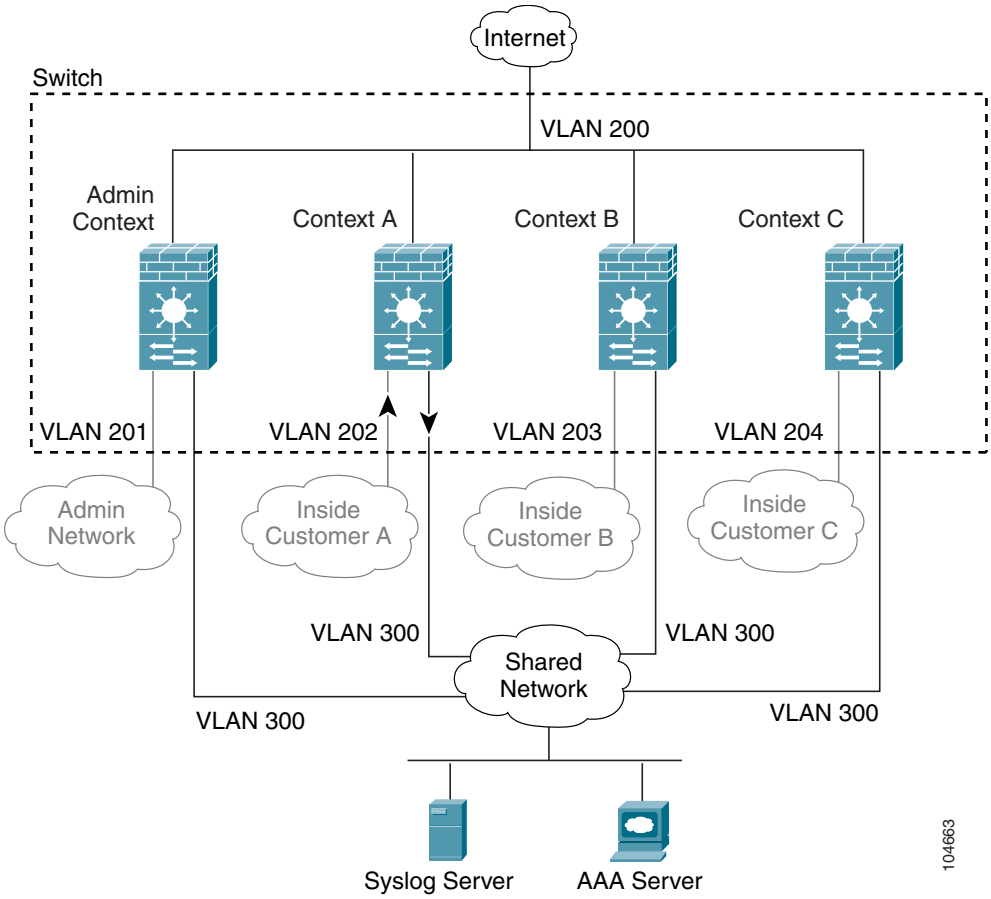
If you put the server on one context network, allow access to the server by authorized users. The benefit of placing the shared resources on one context is that you only need to configure that one context for the shared resources network. The downside is that you must allow outside access to the shared network for the other contexts. Also, because traffic must go out of one context and then back in another, the FWSM has a slightly greater load than if the traffic stays within a context (see Figure 5-4).

Figure 5-4 Shared Resources on One Context



Alternatively, you can share a VLAN on the inside of each context and place the shared resources on a DMZ, labeled "VLAN 300" in Figure 5-5. The downside of placing the shared network inside each context is that you must configure the interface for all contexts; however, this task can be simplified by cutting and pasting between context configurations, and changing only the interface IP address. You also need to make sure that traffic cannot go from one context to another, using the shared network as an interim hop. For example, you could disallow any traffic from originating on the shared network. If you need to originate traffic on the shared interface, for example, to access the Internet, then refer to the "Shared Interface Limitations" section.

Figure 5-5 Shared Resources on a Shared DMZ



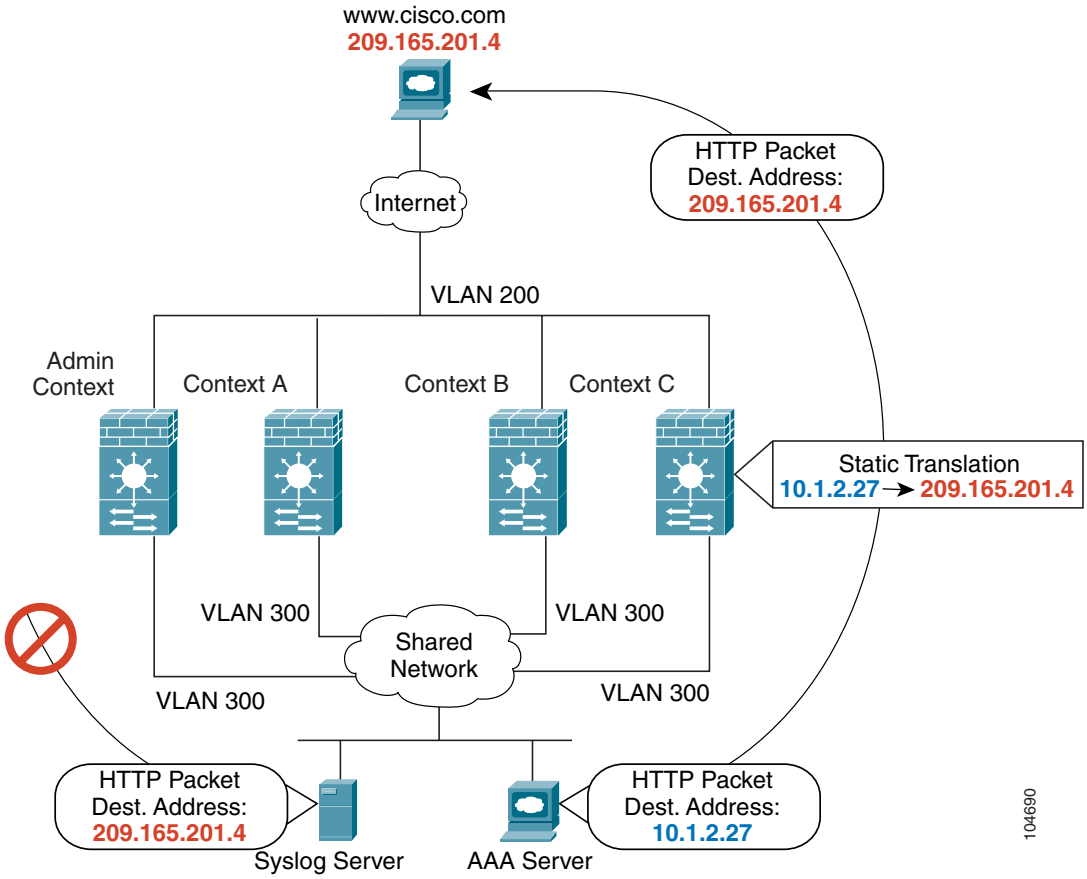
Shared Interface Limitations

For traffic originating on a shared interface, you must configure a static NAT statement for the destination address within a context. This requirement is valid for accessing both higher security interfaces (outside to inside, where a static NAT translation is already required) as well as lower security interfaces (inside to outside), such as connecting to the Internet. This requirement exists because the FWSM classifier must use a unique IP address to determine to which context to send traffic (when you use a shared VLAN, the classifier cannot use the VLAN to classify traffic). However, the FWSM classifier only "knows" about context addresses from already existing NAT translations (returning traffic) and from static NAT translations.

Note You cannot initiate connections from a shared interface when you use NAT exemption for the destination address. The classifier only looks at static statements where the global interface matches the source interface of the packet. Because NAT exemption does not identify a global interface, the classifier does not consider those NAT statements for classification purposes.

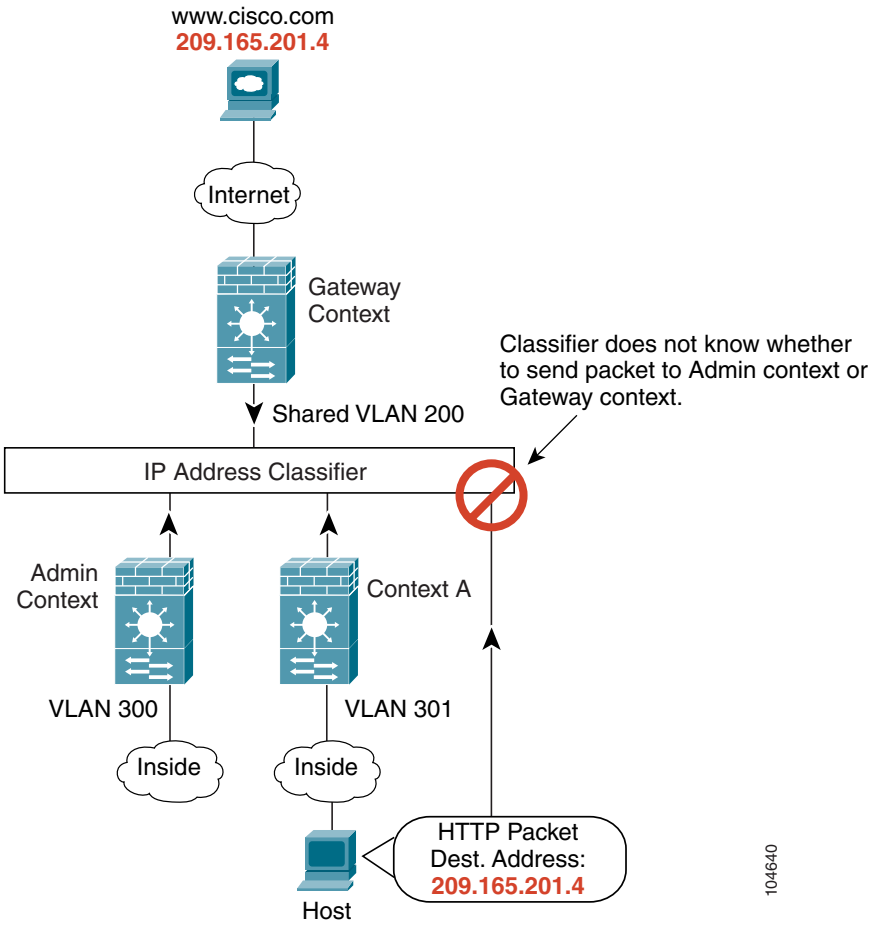
For example, if you send a packet from a host on an inside shared VLAN to www.cisco.com, the FWSM does not know to which context to send the packet unless you statically translate the www.cisco.com IP address in one of the contexts. Figure 5-6 shows two servers on a shared VLAN. One server sends the packet to the translated address, and the FWSM classifies the packet to go through Context C, which includes a static translation for the address. The other server sends the packet to the real untranslated address, and the packet is dropped because the FWSM cannot classify it. If you intend to statically translate addresses for servers like www.cisco.com, then you also need to consider DNS entry addresses and how NAT affects them. For example, if a server sends a packet to www.cisco.com, then the DNS server needs to return the translated address. Managing DNS entries for translated addresses depends on where the DNS server resides. See the "DNS and NAT" section on page 9-13 for more information.

Figure 5-6 Originating Traffic on a Shared VLAN



Because of the limitation for originating traffic on a shared VLAN, a scenario where you place one context behind another is not practical because you would have to configure static statements in the top context for every single outside address that users connected to the bottom context want to access (see Figure 5-7).

Figure 5-7 Cascading Context Limitations



Logging into the FWSM in Multiple Context Mode

When you session into the FWSM, you access the system execution space. If you later configure Telnet or SSH access to a context, you can log into a specific context. If you log into a specific context, you can only access the configuration for that context. However, if you log into the admin context or session into the system execution space, you can access all contexts.

When you change to a context from admin, you continue to use the username and command authorization settings set in the admin context.

The system execution space does not support any AAA commands, but you can configure its own login and enable passwords, as well as usernames in the local database to provide individual logins.

Enabling or Disabling Multiple Context Mode

Your FWSM might already be configured for multiple security contexts depending on how you ordered it from Cisco. If you are upgrading, however, you might need to convert from single mode to multiple mode by following the procedures in this section. To view the mode, enter **show mode**.

The default software license lets you create and use two contexts in addition to the admin context. For more contexts (up to 100), purchase a license from Cisco Systems.

This section includes:

Backing Up the Single Mode Configuration

When you convert from single mode to multiple mode, the FWSM converts the running configuration into two files: a new startup configuration (in Flash) that comprises the system configuration, and admin.cfg (in the disk partition) that comprises the admin context. The original running configuration is saved as old_running.cfg (in disk). The original startup configuration is not saved, so if it differs from the running configuration, you should back it up before proceeding.

Entering an Activation Key for Multiple Security Contexts

The activation key to enable more than two contexts (plus the admin context) is based on your FWSM serial number. Enter the following commands to view your serial number and to enter a key.

- To show the serial number to give to Cisco when ordering your key, enter the following command:

  ```
  FWSM> show version | include Number
  ```

  Enter the pipe character (|) as part of the command.

- To enter the activation key, enter the following command:

  ```
  FWSM(config)# activation-key key
  ```

  The *key* is a four-element hexadecimal string with one space between each element. For example, a key in the correct form might look like the following key:

  ```
  0xe02888da 0x4ba7bed6 0xf1c123ae 0xffd8624e
  ```

  The leading 0x specifier is optional; all values are assumed to be hexadecimal.

  If you are already in multiple context mode, enter this command in the system execution space.

Note The activation key is not stored in your configuration file. The key is tied to the serial number of the device.

Enabling Multiple Context Mode

The context mode (single or multiple) is not stored in the configuration file, even though it does endure reboots. If you need to copy your configuration to another device, you will need to reenter this command on the new device.

When you convert from single mode to multiple mode, the FWSM converts the running configuration into two files: a new startup.cfg (in Flash) that comprises the system configuration, and admin.cfg (in the disk partition) that comprises the admin context. The original running configuration is saved as old_running.cfg (in disk). The original startup configuration is not saved. The FWSM automatically adds an entry for the admin context to the system configuration with the name "admin."

To enable multiple mode, enter the following command:

```
FWSM(config)# mode multiple
```

You are prompted to reboot the FWSM.

Restoring Single Context Mode

If you convert from multiple mode to single mode, the startup configuration is not automatically converted back to the original running configuration. You must copy the backup version of the original running configuration to the current startup configuration. (If you do not have the original configuration, you can start over at the command line.) Because the system configuration does not have any network interfaces as part of its configuration, you must session into the FWSM from the switch to perform the copy (see the "Sessioning and Logging into the Firewall Services Module" section on page 3-1).

To copy the old running configuration to the startup configuration and to change the mode to single mode, enter these commands in the system execution space:

Step 1 To copy the backup version of your original running configuration to the current startup configuration, enter the following command in the system execution space:

```
FWSM(config)# copy disk:old_running.cfg startup-config
```

Step 2 To set the mode to single mode, enter the following command in the system execution space:

```
FWSM(config)# mode single
```

The FWSM reboots.

Configuring Resource Management

By default, all security contexts have unlimited access the resources of the FWSM, except where maximum limits per context are enforced. However, if you find that one or more contexts use too many resources, and they cause other contexts to be denied connections, for example, then you can configure resource management to limit the use of resources per context.

Note The FWSM does not limit the bandwidth per context; however, the switch containing the FWSM can limit bandwidth per VLAN. See the switch documentation for more information.

This section includes the following topics:

Classes and Class Members Overview

The FWSM manages resources by assigning contexts to resource classes. Each context uses the resource limits set by the class. This section includes the following topics:

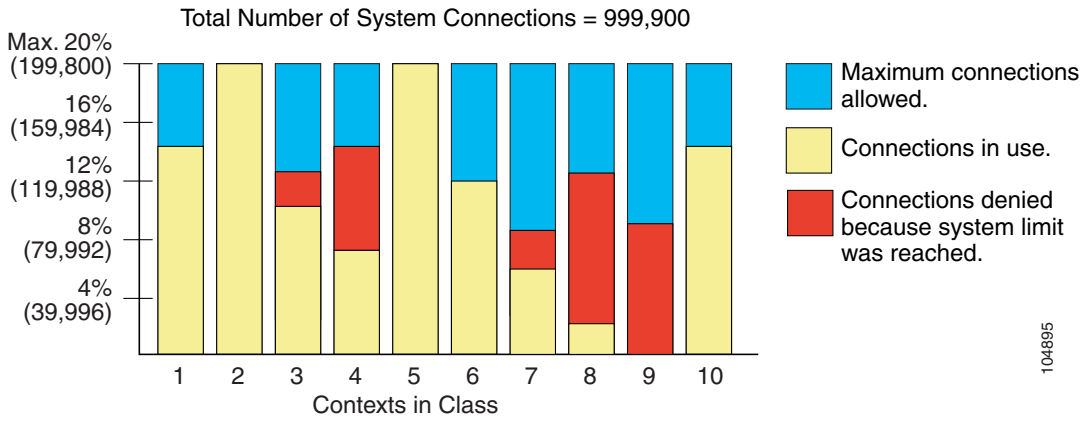
Resource Limits

When you create a class, the FWSM does not set aside a portion of the resources for each context assigned to the class; rather, the FWSM sets the maximum limit for a context. If you oversubscribe resources, or allow some resources to be unlimited, a few contexts can "use up" those resources, potentially affecting service to other contexts.

You can set the limit for all resources together as a percentage of the total available for the device. Also, you can set the limit for individual resources as a percentage or as an absolute value.

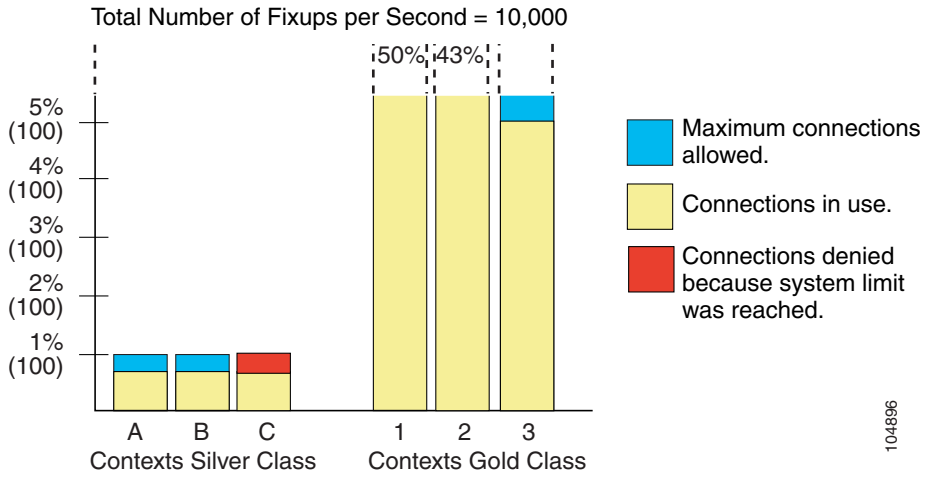
You can oversubscribe the FWSM by assigning more than 100% of the resources across all contexts. For example, you can set the Bronze class to limit connections to 20% per context, and then assign 10 contexts to the class for a total of 200%. If contexts concurrently use more than the system limit, then each context gets less than the 20% you intended (see Figure 5-8).

Figure 5-8 Resource Oversubscription



The FWSM lets you assign unlimited access to one or more resources in a class, instead of a percentage or absolute number. When a resource is unlimited, contexts can use as much of the resource as the system has available. For example, Context A, B, and C are in the Silver Class, which limits each class member to 1% of the system inspections per second, for a total of 3%; but the three contexts are currently only using 2% combined. Gold Class has unlimited access to inspections. The contexts in Gold Class can use more than the 97% of "unassigned" inspections; they can also use the 1% of inspections not currently in use by Context A, B, and C, even if that means that Context A, B, and C are unable to reach their 3% combined limit (see Figure 5-9). Setting unlimited access is similar to oversubscribing the FWSM, except that you have less control over how much you oversubscribe the system.

Figure 5-9 Unlimited Resources



Default Class

All contexts belong to the default class if they are not assigned to another class; you do not have to actively assign a context to the default class.

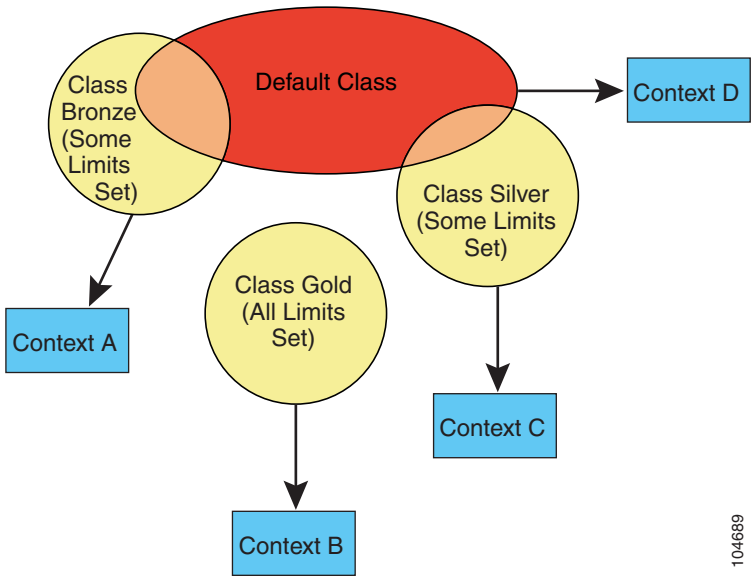
If a context belongs to a class other than the default class, those class settings always override the default class settings. However, if the other class has any settings that are not defined, then the member context uses the default class for those limits. For example, if you create a class with a 2% limit for all concurrent connections, but no other limits, then all other limits are inherited from the default class. Conversely, if you create a class with a 2% limit for *all* resources, the class uses no settings from the default class.

By default, the default class provides unlimited access to resources for all contexts, except for the following limits, which are by default set to the maximum allowed per context:

- Telnet sessions—5 sessions.
- SSH sessions—5 sessions.
- IPSec sessions—5 sessions.
- MAC addresses—65,535 entries.

Figure 5-10 shows the relationship between the default class and other classes. Contexts A and C belong to classes with some limits set; other limits are inherited from the default class. Context B inherits no limits from default because all limits are set in its class, the Gold class. Context D was not assigned to a class, and is by default a member of the default class.

Figure 5-10 Resource Classes



Class Members

To use the settings of a class, assign the context to the class when you define the context. All contexts belong to the default class if they are not assigned to another class; you do not have to actively assign a context to default. You can only assign a context to one resource class. The exception to this rule is that limits that are undefined in the member class are inherited from the default class; so in effect, a context could be a member of default plus another class.

Configuring a Class

To add or change a class in the system configuration, follow these steps. After you add the class, you can add more limits as required by following this procedure again for the same class name and specifying additional limits. You do not need to reenter existing resource commands; the commands you already set remain in place unless you remove them with the **no** form of the command. You can change the value of a particular resource limit by reentering the command with a new value.

To configure a resource class, follow these steps:

Step 1 To specify the class name and enter the class configuration mode, enter the following command in the system execution space:

```
FWSM(config)# class name
```

The *name* is a string up to 20 characters long. To set the limits for the default class, enter **default** for the name.

Step 2 To set the resource limits, see the following options:

- To set all resource limits (shown in Table 5-1), enter the following command:

```
FWSM(config-resmgmt)# limit-resource all {number% | 0}
```

The *number* is an integer greater than or equal to 1. **0** (without a percent sign (%)) sets the resources to unlimited. You can assign more than 100% if you want to oversubscribe the device.

- To set a particular resource limit, enter the following command:

```
FWSM(config-resmgmt)# limit-resource [rate] resource_name number[%]
```

For this particular resource, the limit overrides the limit set for **all**. Enter the **rate** argument to **s**et the rate per second for certain resources. See Table 5-1 for resources for which you can set the rate per second.

Table 5-1 lists the resource types and the limits. See also the **show resource types** command.

Table 5-1 Resource Names and Limits

Resource Name	Minimum and Maximum Number per Context	Total Number for System	Description
mac-addresses	N/A	65 K concurrent	For transparent firewall mode, the number of MAC addresses allowed in the MAC address table.
conns	N/A	999,900 concurrent 102,400 per second (rate)	TCP or UDP connections between any two hosts, including connections between one host and multiple other hosts. **Note** For concurrent connections, the FWSM allocates half of the limit to each of two network processors (NPs) that accept connections. Typically, the connections are divided evenly between the NPs. However, in some circumstances, the connections are not evenly divided, and you might reach the maximum connection limit on one NP before reaching the maximum on the other. In this case, the maximum connections allowed is less than the limit you set. The NP distribution is controlled by the switch based on an algorithm. You can adjust this algorithm on the switch (see the "Customizing the FWSM Internal Interface" section on page 2-11), or you can adjust the connection limit upward to account for the inequity.
fixups	N/A	10,000 per second (rate)	Application inspection.
hosts	N/A	256 K concurrent	Hosts that can connect through the FWSM.
ipsec	1 minimum 5 maximum concurrent	10 concurrent	IPSec sessions
ssh	1 minimum 5 maximum concurrent	100 concurrent	SSH sessions.
syslogs	N/A	30,000 per second (rate)	System messages. **Note** The FWSM can support 30,000 messages per second for messages sent to the FWSM terminal or buffer. If you send messages to a syslog server, the FWSM supports 25,000 per second.
telnet	1 minimum 5 maximum concurrent	100 concurrent	Telnet sessions.
xlates	N/A	256 K concurrent	NAT translations.

For example, to set the default class limit for conns to 10% instead of unlimited, enter the following commands:

```
FWSM(config)# class default
FWSM(config-class)# limit-resource conns 10%
```

All other resources remain at unlimited.

To add a class called gold with all resources set to 5%, except for fixups, with a setting of 10%, enter the following commands:

```
FWSM(config)# class gold
FWSM(config-class)# limit-resource all 5%
FWSM(config-class)# limit-resource fixups 10%
```

To add a class called silver with all resources set to 3%, except for syslogs, with a setting of 500 per second, enter the following commands:

```
FWSM(config)# class silver
FWSM(config-class)# limit-resource all 3%
FWSM(config-class)# limit-resource rate syslogs 500
```

Configuring a Security Context

The security context definition in the system configuration identifies the context name, configuration file URL, VLANs that a context can use, and the resource class to which a context belongs. After you add the context, you can add more VLAN interfaces as required by following this procedure again and specifying additional interfaces. You do not need to reenter other context commands again; the commands you already set remain in place unless you remove them with the **no** form of the command. You can change the value of single-instance commands by reentering the command with a new value. For commands that you can enter multiple times, such as the **allocate-interface** command, you must remove the command with the **no** form and then re-add the altered version.

Note If you do not have an admin context (for example, if you clear the configuration) then the first context you add must be the admin context. Before continuing with this procedure to add a context, enter the following command:

```
FWSM(config)# admin-context name
```

You can now enter the **context** *name* command to match the name you specified for the admin context.

To add or change a context in the system configuration, follow these steps:

Step 1 To add or modify a context, enter the following command in the system execution space:

```
FWSM(config)# context name
```

The *name* is a string up to 32 characters long. This name is case sensitive, so you can have two contexts named “customerA” and “CustomerA,” for example.

We recommend you do not use the names “count” or “detail.” These names are options in the **show context** command, so you cannot use the **show context** command to show information about a context called “count” or “detail.” “system” is a reserved name, and cannot be used.

Step 2 (Optional) To add a description for this context, enter the following command:

```
FWSM(config-context)# description text
```

Step 3 To specify the VLAN interfaces you can use in the context, enter the following command:

```
FWSM(config-context)# allocate-interface vlannumber[-vlannumber] [map_name[-map_name]]
```

You can enter this command multiple times to specify different ranges. For transparent firewall mode, you can only use two interfaces per context.

Enter a VLAN number or a range of VLANs, typically from 1 to 1000 and from 1025 to 4094 (see the switch documentation for supported VLANs). You can assign the same VLANs to multiple contexts, if desired. See the "Sharing Resources and Interfaces Between Contexts" section on page 5-5 for more information about shared VLAN limitations.

The *map_name* is an alphanumeric alias for the VLAN interface that can be used within the context instead of the VLAN number. If you do not specify a mapped name, the VLAN number is used within the context. For security purposes, you might not want the context administrator to know which VLANs are being used by the context. Instead of using the VLAN number in the **nameif** command, for example, you can use the mapped name.

A mapped name must start with a letter, end with a letter or digit, and have as interior characters only letters, digits, or an underscore. For example, you can use the following names:

```
int0
inta
int_0
```

If you specify a range of VLAN IDs, you can specify a matching range of mapped names. Follow these guidelines for ranges:

- The mapped name must consist of an alphabetic portion followed by a numeric portion. The alphabetic portion of the mapped name must match for both ends of the range. For example, enter the following range:

  ```
  int0-int10
  ```

- The numeric portion of the mapped name must include the same quantity of numbers as the **vlan***x*-**vlan***y* statement. For example, both ranges include 100 interfaces:

  ```
  vlan100-vlan199 int1-int100
  ```

  If you enter **vlan100-vlan199 int1-int15** or **vlan100-vlan199 happy1-sad5**, for example, the command fails.

The following example shows VLANs 100, 200, and 300 through 305 assigned to the context. The mapped names are int1 through int8.

```
FWSM(config-context)# allocate-interface vlan100 int1
FWSM(config-context)# allocate-interface vlan200 int2
FWSM(config-context)# allocate-interface vlan300-vlan305 int3-int8
```

Step 4 To identify the URL from which the system downloads the context configuration, enter the following command:

```
FWSM(config-context)# config-url url
```

When you add a context URL, the system immediately loads the context so that it is running.

Note Enter the **allocate-interface** command(s) before you enter the **config-url** command. The FWSM must assign VLAN interfaces to the context before it loads the context configuration; the context configuration might include commands that refer to interfaces (**nameif**, **nat**, **global**...). If you enter the **config-url** command first, the FWSM loads the context configuration immediately. If the context contains any commands that refer to interfaces, those commands fail.

See the following URL syntax:

- **disk://**[*path*/]*filename*
- **ftp://[***user*[**:***password*]@**]***server*/[*path*/]*filename*
- **tftp://***server*/[*path*/]*filename*
- **http://***server*/[*path*/]*filename*
- **https://***server*/[*path*/]*filename*

The FWSM can download a context from a TFTP or FTP server, HTTP or HTTPS server, or from the local disk (called **disk**). The disk is a 64-MB partition of Flash that uses a navigable file system. The disk partition is used only for context storage. The system configuration and the software image reside in the Flash partition (called **flash**).

The server must be accessible from the admin context. The admin context file must be stored on the disk.

The filename does not require a file extension, although we recommend using ".cfg".

If the system cannot retrieve the context configuration file because the server is unavailable, or the file does not yet exist, the system creates a blank context that is ready for you to configure with the command-line interface.

For example, enter the following command:

```
FWSM(config-context)# config-url ftp://joe:passw0rd1@10.1.1.1/configlets/test.cfg
```

Step 5 (Optional) To assign the context to a resource class, enter the following command:

```
FWSM(config-context)# member class_name
```

If you do not specify a class, the context belongs to the default class. You can only assign a context to one resource class.

For example, to assign the context to the gold class, enter the following command:

```
FWSM(config-context)# member gold
```

See the following sample context configurations:

```
FWSM(config)# context administrator
FWSM(config-context)# allocate-interface vlan10
FWSM(config-context)# allocate-interface vlan11
FWSM(config-context)# config-url disk://admin.cfg
FWSM(config-context)# context test
FWSM(config-context)# allocate-interface vlan100 int1
FWSM(config-context)# allocate-interface vlan200 int2
FWSM(config-context)# allocate-interface vlan300-vlan305 int3-int8
FWSM(config-context)# config-url ftp://joe:passw0rd@10.1.1.1/configlets/test.cfg
FWSM(config-context)# member gold
FWSM(config-context)# context sample
FWSM(config-context)# allocate-interface vlan101 int1
FWSM(config-context)# allocate-interface vlan201 int2
```

```
FWSM(config-context)# allocate-interface vlan306-vlan311 int3-int8
FWSM(config-context)# config-url ftp://joe:passw0rd@10.1.1.1/configlets/sample.cfg
FWSM(config-context)# member silver
```

Removing a Security Context

You can only remove a context by editing the system configuration. You cannot remove the current admin context, unless you remove all contexts.

Note If you use failover, there is a delay between when you remove the context on the active unit and when the context is removed on the standby unit. You might see an error message indicating that the number of interfaces on the active and standby units are not consistent; this error is temporary and can be ignored.

See the following commands for removing contexts:

- To remove a single context, enter the following command in the system execution space:

  ```
  FWSM(config)# no context name
  ```

  All context subcommands are also removed.

- To remove all contexts (including the admin context), enter the following command in the system execution space:

  ```
  FWSM(config)# clear context
  ```

Changing the Admin Context

You can set any context to be the admin context.

To set the admin context, enter the following command in the system execution space:

```
FWSM(config)# admin-context context_name
```

Changing Between Contexts and the System Execution Space

If you log into the system execution space or the admin context, you can change between contexts and perform configuration and monitoring tasks within each context. The “running” configuration that you edit in configuration mode, or that is used in the **copy** or **write** commands, depends on your location. When you are in the system execution space, the running configuration consists only of the system configuration; when you are in a context, the running configuration consists only of that context. For example, you cannot view all running configurations (system plus all contexts) by entering the **show running-config** command. Only the current configuration displays.

To change between the system execution space and a context, or between contexts, see the following commands:

- To change to a context, enter the following command:

  ```
  FWSM# changeto context name
  ```

  The prompt changes to the following:

  ```
  FWSM/name#
  ```

- To change to the system execution space, enter the following command:

  ```
  FWSM/admin# changeto system
  ```

  The prompt changes to the following:

  ```
  FWSM#
  ```

Changing the Security Context URL

You cannot change the security context URL without reloading the configuration from the new URL. When you reload the configuration, the new configuration merges with the one in running memory. A merge adds any new commands from the new configuration to the running configuration. If the configurations are the same, no changes occur. If commands conflict or if commands affect the running of the context, then the effect of the merge depends on the command. You might get errors, or you might have unexpected results. If the running configuration is blank (for example, if the server was unavailable and the configuration was never downloaded), then the new configuration is used.

If you do not want to merge the configurations, you can clear the running configuration, which disrupts any communications through the context, and then reload the configuration from the new URL.

To change the URL for a context, follow these steps:

Step 1 If you do not want to merge the configuration, change to the context and clear its configuration by entering the following commands. If you want to perform a merge, skip to step 2.

```
FWSM# changeto context name
FWSM/name# configure terminal
FWSM/name(config)# clear config all
```

Step 2 If required, change to the system execution space by entering the following command:

```
FWSM/name(config)# changeto system
```

Step 3 To enter the context configuration mode for the context you want to change, enter the following command:

```
FWSM(config)# context name
```

Step 4 To enter the new URL, enter the following command:

```
FWSM(config)# config-url new_url
```

The system immediately loads the context so that it is running.

Reloading a Security Context

You can reload the context in two ways:

- Clear the running configuration and then import the startup configuration.

  This action clears most attributes associated with the context, such as connections, and NAT tables.

- Remove the context from the system configuration.

  This action clears additional attributes, such as memory allocation, which might be useful for troubleshooting. However, to add the context back to the system requires you to respecify the URL, VLANs, and class membership.

This section includes the following topics:

Reloading by Clearing the Configuration

To reload the context by clearing the context configuration, and reloading the configuration from the URL, follow these steps:

Step 1 To change to the context that you want to reload, enter the following command:

```
FWSM# changeto context name
```

Step 2 To access configuration mode, enter the following command:

```
FWSM/name# configure terminal
```

Step 3 To clear the running configuration, enter the following command:

```
FWSM/name(config)# clear configure all
```

This command stops the context from running.

Step 4 To reload the configuration, enter the following command:

```
FWSM/name(config)# copy startup-config running-config
```

The FWSM copies the configuration from the URL specified in the system configuration. You cannot change the URL from within a context.

Reloading by Removing and Re-adding the Context

To reload the context by removing the context and then re-adding it, follow the steps in the following sections:

1. "Removing a Security Context" section on page 5-20
2. "Configuring a Security Context" section on page 5-17

Monitoring Security Contexts

This section describes how to view and monitor context information, and includes the following topics:

Viewing Context Information

From the system execution space, you can view a list of contexts including the name, class, interfaces, and configuration file URL.

From the system execution space, view all contexts by entering the following command:

```
FWSM# show context [name [detail]| count]
```

The **detail** option shows additional information. See the sample displays below for more information.

If you want to show information for a particular context, specify the name.

The **count** option shows the total number of contexts.

The following sample display shows three contexts:

```
FWSM# show context
Context Name      Class      Interfaces           URL
*admin            default    Vlan10,22,55-57      disk:/admin.cfg
 contexta         gold       vlan10,100-101       disk:/contexta.cfg
 contextb         silver     vlan10,110-111       disk:/contextb.cfg

Total active Security Contexts: 3
```

Table 5-2 shows each field description.

Table 5-2 show context Fields

Field	Description
Context Name	Lists all context names. The context name with the asterisk (*) is the admin context.
Class	The class to which the context belongs.
Interfaces	The VLAN interfaces assigned to the context.
URL	The URL from which the FWSM loads the context configuration.

The following sample display shows the **detail** option:

```
FWSM# show context detail
Context "admin", is ADMIN and active
 Config URL: disk:/admin.cfg
 Interfaces: Vlan10,22,55-57
 Class: default, Flags: 0x00000057, ID: 1
```

```
Context "contexta", is active
 Config URL: disk:/contexta.cfg
 Interfaces: vlan10,100-101
 Class: default, Flags: 0x00000055, ID: 2
```

The “Flags” and “ID” fields are for internal use only.

Viewing Resource Allocation

From the system execution space, you can view the allocation for each resource across all classes and class members.

From the system execution space, view the resource allocation by entering the following command:

```
FWSM# show resource allocation [detail]
```

This command shows the resource allocation, but does not show the actual resources being used. See the “Viewing Resource Usage” section on page 5-26 for more information about actual resource usage.

The **detail** argument shows additional information. See the sample displays below for more information.

The following sample display shows the total allocation of each resource as an absolute value and as a percentage of the available system resources:

```
FWSM# show resource allocation
Resource                    Total       % of Avail
 Conns [rate]                35000           35.00%
 Fixups [rate]               35000           35.00%
 Syslogs [rate]              10500           35.00%
 Conns                      305000           30.50%
 Hosts                       78842           30.07%
 IPsec                           7           35.00%
 SSH                            35           35.00%
 Telnet                         35           35.00%
 Xlates                      91749           34.99%
 All                     unlimited
```

Table 5-3 shows each field description.

Table 5-3 show resource allocation Fields

Field	Description
Resource	The name of the resource that you can limit. See the “Configuring a Class” section on page 5-14 for more information about each resource name.
Total	The total amount of the resource that is allocated across all contexts. The amount is an absolute number of concurrent instances or instances per second. If you specified a percentage in the class definition, the FWSM converts the percentage to an absolute number for this display.
% of Avail	The percentage of the total system resources that is allocated across all contexts.

The following sample display shows the **detail** option:

```
FWSM# show resource allocation detail
Resource Origin:
    A    Value was derived from the resource 'all'
    C    Value set in the definition of this class
    D    Value set in default class
Resource         Class          Mmbrs  Origin      Limit      Total    Total %
Conns [rate]     default          all      CA  unlimited
                 gold               1       C      34000      34000     20.00%
                 silver             1      CA      17000      17000     10.00%
                 bronze             0      CA       8500
                 All Contexts:      3                         51000     30.00%

Fixups [rate]    default          all      CA  unlimited
                 gold               1      DA  unlimited
                 silver             1      CA      10000      10000     10.00%
                 bronze             0      CA       5000
                 All Contexts:      3                         10000     10.00%

Syslogs [rate]   default          all      CA  unlimited
                 gold               1       C       6000       6000     20.00%
                 silver             1      CA       3000       3000     10.00%
                 bronze             0      CA       1500
                 All Contexts:      3                          9000     30.00%

Conns            default          all      CA  unlimited
                 gold               1       C     200000     200000     20.00%
                 silver             1      CA     100000     100000     10.00%
                 bronze             0      CA      50000
                 All Contexts:      3                        300000     30.00%

Hosts            default          all      CA  unlimited
                 gold               1      DA  unlimited
                 silver             1      CA      26214      26214      9.99%
                 bronze             0      CA      13107
                 All Contexts:      3                         26214      9.99%

IPSec            default          all       C          5
                 gold               1       D          5          5     50.00%
                 silver             1      CA          1          1     10.00%
                 bronze             0      CA  unlimited
                 All Contexts:      3                            11    110.00%

SSH              default          all       C          5
                 gold               1       D          5          5      5.00%
                 silver             1      CA         10         10     10.00%
                 bronze             0      CA          5
                 All Contexts:      3                            20     20.00%

Telnet           default          all       C          5
                 gold               1       D          5          5      5.00%
                 silver             1      CA         10         10     10.00%
                 bronze             0      CA          5
                 All Contexts:      3                            20     20.00%

Xlates           default          all      CA  unlimited
                 gold               1      DA  unlimited
                 silver             1      CA      23040      23040     10.00%
                 bronze             0      CA      11520
                 All Contexts:      3                         23040     10.00%

mac-addresses    default          all       C      65535
                 gold               1       D      65535      65535    100.00%
                 silver             1      CA       6553       6553      9.99%
                 bronze             0      CA       3276
                 All Contexts:      3                        137623    209.99%
```

Table 5-4 shows each field description.

Table 5-4 show resource allocation detail Fields

Field	Description
Resource	The name of the resource that you can limit. See the "Configuring a Class" section on page 5-14 for more information about each resource name.
Class	The name of each class, including the default class. The All contexts field shows the total values across all classes.
Mmbrs	The number of contexts assigned to each class.
Origin	The origin of the resource limit, as follows: • A—You set this limit with the **all** option, instead of as an individual resource. • C—This limit is derived from the member class. • D—This limit was not defined in the member class, but was derived from the default class. For a context assigned to the default class, the value will be "C" instead of "D." The FWSM can combine "A" with "C" or "D."
Limit	The limit of the resource per context, as an absolute number. If you specified a percentage in the class definition, the FWSM converts the percentage to an absolute number for this display.
Total	The total amount of the resource that is allocated across all contexts in the class. The amount is an absolute number of concurrent instances or instances per second. If the resource is unlimited, this display is blank.
% of Avail	The percentage of the total system resources that is allocated across all contexts in the class. If the resource is unlimited, this display is blank.

Viewing Resource Usage

From the system execution space, you can view the resource usage.

From the system execution space, view the resource usage for each context by entering the following command:

```
FWSM# show resource usage [context context_name | top n | all | summary | system]
[resource {[rate] resource_name | all} | detail] [counter counter_name [count_threshold]]
```

all is the default, and shows resource usage for each context individually.

Enter the **top** *n* keyword to show the contexts that are the top *n* users of the specified resource. You must specify a single resource type, and not **resource all**, with this option.

The **summary** option shows the total for all contexts together. For example, the denied column shows the items that have been denied for each context limit. The **system** option shows the counts for the entire system. For the limit and denied counts, for example, you only see a number in the denied column if the system limit is reached, not if one or more context limits are reached.

Note When the TCP intercept feature intercepts connections (see the embryonic connection limit in the **nat** and **static** commands), the FWSM includes these connections only in the system counts and not in individual context counts. To see the intercepted connections separate from other connections, use the **system detail** option.

For the resource name, see Table 5-1 on page 5-16 for resource names.

The **detail** keyword shows the resources you can limit in a class, plus other system resources for which you cannot configure limits.

The **counter** *counter_name* is one of the following keywords:

- **current**—Shows the active concurrent instances or the current rate of the resource.
- **peak**—Shows the peak concurrent instances, or the peak rate of the resource since the statistics were last cleared, either using the **clear resource usage** command or because the device rebooted.
- **denied**—Shows the number of denied uses of the resource, since the resource statistics were last cleared.
- **all**—(Default) Shows all statistics.

The *count_threshold* sets the number above which resources are shown. The default is 1. If the usage of the resource is below the number you set, then the resource is not shown. If you specify **all** for the counter name, then the *count_threshold* applies to the current usage.

Note To show all resources, set the *count_threshold* to **0**.

The following sample display shows the resource usage for all contexts and all resources.

```
FWSM# show resource usage summary
Resource              Current         Peak      Limit     Denied Context
Syslogs [rate]           1743         2132      12000(U)       0 Summary
Conns                     584          763     100000(S)       0 Summary
Xlates                   8526         8966      93400          0 Summary
Hosts                     254          254     262144          0 Summary
Conns [rate]              270          535      42200       1704 Summary
Fixups [rate]             270          535     100000(S)       0 Summary
U = Some contexts are unlimited and are not included in the total.
S = All contexts are unlimited; system limit is shown.
```

Configuring Basic Settings

This chapter tells how to configure basic settings on your Firewall Services Module (FWSM). This chapter includes the following sections:

Changing the Passwords

This section tells how to change the login password and enable password from their default settings, as well as how to change the maintenance partition passwords. The maintenance partition is used for troubleshooting, recovering from a corrupted image, or recovering lost passwords. See the following topics:

Note In multiple context mode, every context and the system execution space has its own login policies and passwords.

Changing the Login Password

By default, the login password is "cisco."

To change the password, enter the following command in privileged mode:

```
FWSM/contexta(config)# {passwd | password} password
```

You can enter **passwd** or **password**. The *password* is a case-sensitive password of up to 16 alphanumeric and special characters. You can use any character in the password except a question mark or a space.

The password is saved in the configuration in encrypted form, so you cannot view the original password after you enter it. Use the **clear password** command to restore the password to the default setting.

Changing the Enable Password

By default, the enable password is blank.

To change enable the password, enter the following command in privileged mode:

```
FWSM/contexta(config)# enable password password
```

The *password* is a case-sensitive password of up to 16 alphanumeric and special characters. You can use any character in the password except a question mark or a space.

This command changes the password for the highest privilege level. If you configure local command authorization, you can set enable passwords for each privilege level from 0 to 15. See the "Configuring Command Authorization" section on page 12-10 for more information.

The password is saved in the configuration in encrypted form, so you cannot view the original password after you enter it. Enter the **enable password** command without a password to set the password to the default, which is blank.

Changing the Maintenance Partition Passwords

The maintenance partition is valuable for troubleshooting. For example, you can install new software to an application partition, reset passwords, or show crash dump information from the maintenance partition. You can only access the maintenance partition by sessioning into the FWSM (see the "Sessioning and Logging into the Firewall Services Module" section on page 3-1).

The maintenance partition has two user levels with different access privileges:

- root—Lets you configure the network partition parameters, upgrade the software images on the application partitions, change the guest account password, and enable or disable the guest account.

  The default password is "cisco."

- guest—Lets you configure the network partition parameters and show crash dump information.

  The default password is "cisco."

To change the maintenance partition passwords for both users, follow these steps:

Step 1 To reboot the FWSM into the maintenance partition, enter the command for your operating system:

- Cisco IOS software:

```
Router# hw-module module mod_num reset cf:1
```

- Catalyst OS:

```
Console> (enable) reset mod_num boot cf:1
```

Step 2 To session into the FWSM, enter the command for your operating system:

- Cisco IOS software:

```
Router# session slot mod_num processor 1
```

- Catalyst OS:

```
Console> (enable) session mod_num
```

Step 3 Log in as root by entering the following command:

```
Login: root
```

Step 4 Enter the password at the prompt:

```
Password:
```

The default password is “cisco”.

Step 5 Change the root password by entering the following command:

```
root@localhost# passwd
```

Step 6 Enter the new password at the prompt:

```
Changing password for user root
New password:
```

Step 7 Enter the new password again:

```
Retype new password:
passwd: all authentication tokens updated successfully
```

Step 8 Change the guest password by entering the following command:

```
root@localhost# passwd-guest
```

Step 9 Enter the new password at the prompt:

```
Changing password for user guest
New password:
```

Step 10 Enter the new password again:

```
Retype new password:
passwd: all authentication tokens updated successfully
```

This example shows how to set the password for the root account:

```
root@localhost# passwd
Changing password for user root
New password: *sh1p
Retype new password: *sh1p
passwd: all authentication tokens updated successfully
```

This example shows how to set the password for the guest account:

```
root@localhost# passwd-guest
Changing password for user guest
New password: f1rc8t
Retype new password: f1rc8t
passwd: all authentication tokens updated successfully
```

Setting the Host Name

When you set a host name for the FWSM, that name appears in the command line prompt. If you establish sessions to multiple devices, the host name helps you keep track of where you enter commands. By default, the host name is "FWSM".

For multiple context mode, the host name that you set in the system execution space appears in the command line prompt for all contexts.

The host name that you optionally set within a context does not appear in the command line, but is used for RSA key generation. If you do not set a host name within a context, the context name is used as the host name in the certificate. RSA keys are required for SSH, the HTTPS server, and can be used for VPN. You should also set the domain name for RSA key generation (see "Setting the Domain Name"). If you change the host name after you generate keys, you need to regenerate the keys using the **ca generate rsa key** command.

To specify the host name for the FWSM or for a context, enter the following command:

```
FWSM(config)# hostname name
```

This name can be up to 63 characters, including alphanumeric characters, spaces or any of the following special characters: ` ( ) + - , . / : = ?.

In single mode and in the system execution space in multiple mode, this name appears in the command line prompt. For example:

```
FWSM(config)# hostname farscape
farscape(config)#
```

For a context, this name is used for RSA key generation. If you do not set a host name within a context, the context name is used for the host name in the key. You can view a context host name using the **show hostname** command.

Setting the Domain Name

The domain name is used for RSA key generation. RSA keys are required for SSH, the HTTPS server, and can be used for VPN. You should also set the host name for key generation (see "Setting the Host Name"). If you change the domain name after generating keys, you need to regenerate the keys using the **ca generate rsa key** command.

For multiple context mode, you can set the domain name for each context, as well as within the system execution space.

To specify the domain name for the FWSM, enter the following command:

```
FWSM/contexta(config)# domain-name name
```

For example, to set the domain as cisco.com, enter the following command:

```
FWSM(config)# domain-name cisco.com
```

Adding a Login Banner

You can configure a message to display when a user connects to the FWSM, before a user logs in, or before a user enters privileged mode.

To configure a login banner, enter the following command in the system execution space or within a context:

```
FWSM/contexta(config)# banner {exec | login | motd} text
```

Adds a banner to display at one of three times: when a user first connects (message-of-the-day (**motd**)), when a user logs in (**login**), and when a user accesses privileged mode (**exec**). When a user connects to the FWSM, the message-of-the-day banner appears first, followed by the login banner and prompts. After the user successfully logs in to the FWSM, the exec banner displays.

For the banner text, spaces are allowed but tabs cannot be entered using the CLI. You can dynamically add the host name or domain name of the FWSM by including the strings **$(hostname)** and **$(domain)**. If you configure a banner in the system configuration, you can use that banner text within a context by using the **$(system)** string in the context configuration.

To add more than one line, precede each line by the **banner** command.

For example, to add a message-of-the-day banner, enter:

```
FWSM/contexta(config)# banner motd Welcome to the $(hostname) firewall.
FWSM/contexta(config)# banner motd Contact me at admin@admin.com for any
FWSM/contexta(config)# banner motd issues.
```

Configuring Interfaces

By default, all interfaces are enabled. For each interface, you must provide a name and a security level.

Note If you are using failover, do not use this procedure to name interfaces that you are reserving for failover and stateful failover communications. See Chapter 15, "Using Failover," to configure the failover and state links.

This section includes the following topics:

Security Level Overview

Each interface must have a security level from 0 (lowest) to 100 (highest). For example, you should assign your most secure network, such as the inside host network, to level 100. While the outside network connected to the Internet can be level 0. Other networks, such as DMZs can be in between. You can assign interfaces to the same security level. See the "Allowing Communication Between Interfaces on the Same Security Level" section on page 6-8 for more information.

For interfaces that are on different security levels, the level controls the following behavior:

- NAT—When hosts on a higher security interface (inside) access hosts on a lower security interface (outside), you must configure Network Address Translation (NAT) for the inside hosts *or* specifically configure the inside hosts to bypass NAT.

  An inside host can communicate with the untranslated local address of the outside host without any special configuration on the outside interface. However, you can also optionally perform NAT on the outside network.

- Inspection engines—Some inspection engines are dependent on the security level:
  - SMTP inspection engine—Applied only for inbound connections (from lower level to higher level), which protects the SMTP servers on the higher security interface.
  - NetBIOS inspection engine—Applied only for outbound connections.
  - XDMCP inspection engine—The XDMCP server can be configured only on the outside interface.
  - OraServ inspection engine—If a control connection for the OraServ port exists between a pair of hosts, then only an inbound data connection is permitted through the FWSM.
- Filtering—HTTP(S) and FTP filtering applies only for outbound connections (from a higher level to a lower level).
- TCP intercept—The TCP intercept feature only applies to hosts or servers on a higher security level. See the "Other Protection Features" section on page 1-6 for more information about TCP intercept. This feature is configured using the *emb_limit* option in the **nat** and **static** commands.

- TCP sequence randomization—Each TCP connection has two Initial Sequence Numbers (ISNs): one generated by the client and one generated by the server. The FWSM randomizes the ISN that is generated by the host/server on the higher security interface. At least one of the ISNs must be randomly generated so that attackers cannot predict the next ISN and potentially hijack the session.
- Maximum connections limit—You can set a limit on the number of TCP and UDP connections allowed through the FWSM, but only connections from a higher security interface to a lower security interface are tracked. This limit is set using the *max_conns* option in the **nat** and **static** commands.
- **established** command—This command allows return connections from a lower security host to a higher security host if there is already an established connection from the higher level host to the lower level host.

These behaviors do not affect interfaces that are on the same security level. For example, you do not have to perform NAT, nor do you have to configure the interfaces to bypass NAT. You can, however, optionally configure NAT for these interfaces. Similarly, inspection engines are applied to both interfaces, as is filtering.

Note By default, the Cisco PIX firewall allows traffic to flow freely from an inside network (higher security level) to an outside network (lower security level). However, the FWSM does not allow any traffic to pass between interfaces unless you explicitly permit it with an access control list (ACL). While you still have to specify the security level for an interface on the FWSM, the security level does not provide an explicit permission for traffic to travel from a high security interface to a low security interface.

Setting the Name and Security Level

By default, all interfaces are enabled. However, you must assign a name and security level to each interface before you can fully configure the FWSM. Many commands use the interface name instead of the interface (VLAN) ID.

You can assign a name to a VLAN that has not yet been assigned to the FWSM (see the "Assigning VLANs to the Firewall Services Module" section on page 2-2), but you see a warning message.

Note If you are using failover, do not use this procedure to name interfaces that you are reserving for failover and stateful failover communications. See Chapter 15, "Using Failover," to configure the failover and state links.

For multiple context mode, follow these guidelines:

- Configure the context interfaces from within each context.
- You can only configure context interfaces that you already assigned to the context in the system configuration.
- The system configuration does not include configurable interfaces, except for failover interfaces. Do not configure failover interfaces with this procedure. See Chapter 15, "Using Failover," for more information.

In transparent firewall mode, you can use only two interfaces, one inside and one outside.

Note If you change the security level of an interface, and you do not want to wait for existing connections to time out before the new security information is used, you can clear the translation table using the **clear xlate** command. However, clearing the translation table disconnects all current connections.

To name an interface, enter the following command:

```
FWSM/contexta(config)# nameif {vlan n | context_map_name} name [security]n
```

For multiple context mode, if you gave the VLAN interface a mapped name for the context in the system configuration, then you must use the mapped name.

The *name* is a text string up to 48 characters, and is not case-sensitive.

The security level is an integer between 0 and 100. 0 is the least secure and 100 the most secure. You can optionally include the word **security** before the level number to make your configuration easier to read. To assign more than one interface to the same level, see the "Allowing Communication Between Interfaces on the Same Security Level" section on page 6-8 to enable this feature.

For example, enter the **show nameif** command to view the interface names:

```
FWSM# show nameif
nameif vlan100 outside security0
nameif vlan101 inside security100
nameif vlan102 dmz security50
```

Allowing Communication Between Interfaces on the Same Security Level

By default, interfaces on the same security level cannot communicate with each other, even if you configure NAT and ACLs.

Allowing communication between same security interfaces provides the following benefits:

- You do not need to configure NAT between same security interfaces.

  You can, however, configure NAT if desired. If you configure dynamic NAT for an interface, then to allow connections initiated from another interface, even if it is on the same security level, you need to configure static NAT.

  If you want to configure connection limits but do not want to configure NAT (where connection limits are set), you can configure identity NAT or NAT exemption. (See the "Configuring Connection Limits for Non-NAT Configurations" section on page 6-9 or the "Bypassing NAT" section on page 9-29.)

- You can configure more than 101 communicating interfaces.

  If you use different levels for each interface, you can configure only one interface per level (0 to 100).

- You want protection features to be applied equally for traffic between two interfaces; for example, you have two departments that are equally secure.

  For different security level interfaces, many protection features apply only in one direction, for example, inspection engines, TCP intercept, and connection limits.

If you enable same security interface communication, you can still configure interfaces at different security levels as usual.

To enable interfaces on the same security level so that they can communicate with each other, enter the following command:

```
FWSM/contexta(config)# same-security-traffic permit inter-interface
```

To disable this setting, add **no** before the command.

Turning Off and Turning On Interfaces

All interfaces are enabled by default. If you disable or reenable the interface within a context, only that context interface is affected. But if you disable or reenable the interface in the system execution space, then you affect that VLAN interface for all contexts.

To disable an interface or reenable it, follow these steps:

Step 1 To enter the interface configuration mode, enter the following command:

```
FWSM/contexta(config)# interface interface_name
```

Step 2 To disable the interface, enter the following command:

```
FWSM/contexta(config-interface)# shutdown
```

Step 3 To reenable the interface, enter the following command:

```
FWSM/contexta(config-interface)# no shutdown
```

Configuring Connection Limits for Non-NAT Configurations

Transparent firewall mode

Same security level mode

The NAT configuration enables you to set connection limits for traffic. For transparent firewall mode or for same security interfaces on which you do not want to configure NAT (see the "Allowing Communication Between Interfaces on the Same Security Level" section on page 6-8), you can configure identity NAT to set these limits. Identity NAT lets you specify the addresses for which you want to set limits, but no translation is performed. (For same security interfaces, you can configure any method for bypassing NAT, including NAT exemption. See the "Bypassing NAT" section on page 9-29 for more information. For transparent mode, the FWSM supports only the following method.)

To set connection limits for the inside interface (transparent mode) or for any same security interface, enter the following command:

```
FWSM/contexta(config)# static (inside_interface,outside_interface) local_ip_address
local_ip_address netmask mask [norandomseq] [[tcp] tcp_max_conns [emb_limit]]
[udp udp_max_conns]
```

Enter the same IP address for both *local_ip_address* options.

Set one or more of the following options:

- **norandomseq—**No TCP Initial Sequence Number (ISN) randomization. Only use this option if another in-line firewall is also randomizing sequence numbers and the result is scrambling the data. See the "Security Level Overview" section on page 6-6 for information about TCP sequence numbers.
- **tcp** *tcp_max_conns,* **udp** *udp_max_conns*—The maximum number of simultaneous TCP and/or UDP connections for the entire subnet up to 65,536. The default is 0 for both protocols, which means the maximum connections.
- *emb_limit*—The maximum number of embryonic connections per host up to 65,536. An embryonic connection is a connection request that has not finished the necessary handshake between source and destination. This limit enables the TCP Intercept feature. (See the "Other Protection Features" section on page 1-6 for more information.) The default is **0**, which means the maximum embryonic connections. You must enter the **tcp** *tcp_max_conns* before you enter the *emb_limit*. If you want to use the default value for *tcp_max_conns*, but change the *emb_limit*, then enter **0** for *tcp_max_conns*.

For example, to set options for the host 10.1.1.1, enter the following command:

```
FWSM/contexta(config)# static (inside,outside) 10.1.1.1 10.1.1.1 netmask 255.255.255.255
norandomseq tcp 1000 200 udp 1000
```

Configuring Bridging Parameters and ARP Inspection

Transparent firewall mode only

This chapter tells how to customize bridging operations and how to enable ARP inspection for the transparent firewall. This chapter includes the following sections:

Customizing the MAC Address Table

This section includes the following topics:

MAC Address Table Overview

The FWSM learns and builds a MAC address table in a similar way as a normal bridge or switch: when a device sends a packet through the FWSM, the FWSM adds the MAC address to its table. The table associates the MAC address with the source interface so that the FWSM knows to send any packets addressed to the device out the correct interface.

Because the FWSM is a firewall, if the destination MAC address of a packet is not in the table, the FWSM does not flood the original packet on all interfaces as a normal bridge does. Instead, it generates the following packets for directly connected devices or for remote devices:

- Packets for directly connected devices—The FWSM generates an ARP request for the destination IP address, so that the FWSM can learn which interface receives the ARP response.
- Packets for remote devices—The FWSM generates a ping to the destination IP address so that the FWSM can learn which interface receives the ping reply.

The original packet is dropped.

Note For multiple context mode, you can limit the maximum number of mac address table entries in the context class. See the "Configuring a Class" section on page 5-14 for more information.

Adding a Static MAC Address

Normally, MAC addresses are added to the MAC address table dynamically as traffic from a particular MAC address enters an interface. You can add static MAC addresses to the MAC address table if desired.

To add a static MAC address to the MAC address table, enter the following command:

```
FWSM/contexta(config)# mac-address-table static interface_name mac_address
```

The *interface_name* is the source interface.

Setting the MAC Address Timeout

The default timeout value for dynamic MAC address table entries is 5 minutes, but you can change the timeout.

To change the timeout, enter the following command:

```
FWSM/contexta(config)# mac-address-table aging-time timeout_value
```

The *timeout_value* (in minutes) is between 5 and 720 (12 hours). 5 minutes is the default.

Disabling MAC Address Learning

By default, each interface automatically learns the MAC addresses of entering traffic, and the FWSM adds corresponding entries to the MAC address table. You can disable MAC address learning if desired.

To disable MAC address learning, enter the following command:

```
FWSM/contexta(config)# mac-learn interface_name disable
```

The **no** form of this command reenables MAC address learning.

Viewing the MAC Address Table

You can view the entire MAC address table (including static and dynamic entries for both interfaces), or you can view the MAC address table for an interface.

To view the MAC address table, enter the following command:

```
FWSM/contexta# show mac-address-table [interface_name]
```

The following example shows the entire MAC address table:

```
FWSM/contexta# show mac-address-table
interface          mac address       type      Time Left
------------------------------------------------------------------------
outside            0009.7cbe.2100    static      -
inside             0010.7cbe.6101    static      -
inside             0009.7cbe.5101    dynamic     10
```

The following example shows the MAC address table for the inside interface:

```
FWSM/contexta# show mac-address-table inside
interface          mac address       type      Time Left
------------------------------------------------------------------------
inside             0010.7cbe.6101    static      -
inside             0009.7cbe.5101    dynamic     10
```

Configuring ARP Inspection

This section describes ARP inspection and how to enable it, and includes the following topics:

ARP Inspection Overview

By default, ARP inspection is disabled on all interfaces; all ARP packets are allowed through the FWSM.

When you enable ARP inspection, the FWSM compares the MAC address, IP address, and source interface in all ARP packets to static entries in the ARP table, and takes the following actions:

- If the IP address, MAC address, and source interface match an ARP entry, the packet is passed through.
- If there is a mismatch between the MAC address, the IP address, or the interface, then the FWSM drops the packet.
- If the ARP packet does not match any entries in the static ARP table, then you can set the FWSM to either forward the packet out all interfaces (flood), or to drop the packet.

ARP inspection prevents malicious users from impersonating other hosts or routers (known as ARP spoofing). ARP spoofing can enable a “man-in-the-middle” attack. For example, a host sends an ARP request to the gateway router; the gateway router responds with the gateway router MAC address.

The attacker, however, sends another ARP response to the host with the attacker MAC address instead of the router MAC address. The attacker can now intercept all the host traffic before forwarding it on to the router.

ARP inspection ensures that an attacker cannot send an ARP response with the attacker MAC address, so long as the correct MAC address and the associated IP address are in the static ARP table.

Adding a Static ARP Entry

ARP inspection compares ARP packets with static ARP entries in the ARP table.

To add a static ARP entry, enter the following command:

```
FWSM/contexta(config)# arp interface_name ip_address mac_address
```

For example, to allow ARP responses from the router at 10.1.1.1 with the MAC address 0009.7cbe.2100 on the outside interface, enter the following command:

```
FWSM/contexta(config)# arp outside 10.1.1.1 0009.7cbe.2100
```

Enabling ARP Inspection

To enable ARP inspection, enter the following command:

```
FWSM/contexta(config)# arp-inspection interface_name enable [flood | no-flood]
```

Where **flood** (the default) forwards non-matching ARP packets out all interfaces, and **no-flood** drops non-matching packets.

For example, to enable ARP inspection on the outside interface, and to drop all non-matching ARP packets, enter the following command:

```
FWSM/contexta(config)# arp-inspection outside enable no-flood
```

Configuring IP Addresses, Routing, and DHCP

This chapter tells how to configure IP addresses, static routes, dynamic routing, and DHCP on the Firewall Services Module (FWSM).

For routed and transparent mode, see the following sections:

For routed mode only, see the following sections:

Note Multiple security contexts do not support dynamic routing protocols, such as RIP and OSPF.

Configuring IP Addresses

This section describes how to set the IP address(es) for routed mode or for transparent mode, and includes the following topics:

Assigning IP Addresses to Interfaces for a Routed Firewall

Routed firewall mode only

To assign an IP address to a VLAN interface, enter the following command:

```
FWSM/contexta(config)# ip address interface_name ip_address mask [standby ip_address]
```

In single context mode, each interface address must be on a unique subnet. In multiple context mode, if this interface is on a shared VLAN, then the IP address must be unique, and cannot be used by another context on the shared VLAN. If the VLAN is unique, this IP address can be used by other contexts if desired.

The **standby** keyword and address is used for failover. See the "Configuring Failover" section on page 15-14 for more information.

For example, to set the IP address of the inside interface, enter the following command:

```
FWSM/contexta(config)# ip address inside 192.168.1.1 255.255.255.0
```

Setting the Management IP Address for a Transparent Firewall

Transparent firewall mode only

A transparent firewall does not participate in IP routing. The only IP configuration required for the FWSM is to set the management IP address. This address is required because the FWSM uses this address as the source address for traffic originating on the FWSM, such as system messages or communications with AAA servers. You can also use this address for remote management access.

For multiple context mode, set the management IP address within each context.

To set the management IP address, enter the following command:

```
FWSM/contexta(config)# ip address ip_address [mask] [standby ip_address]
```

This address must be on the same subnet as the upstream and downstream routers.

The **standby** keyword and address is used for failover. See the "Configuring Failover" section on page 15-14 for more information.

Configuring the Default Route

The default route identifies the router IP address to which the FWSM sends all IP packets for which it does not have a route. The FWSM might receive a default route from the dynamic routing protocol (single mode only), but we recommend setting a static default route as a backup.

For transparent firewall mode, for traffic that originates on the FWSM and is destined for a non-directly connected network, configure either a default route or static routes so the FWSM knows out of which interface to send traffic. Traffic that originates on the FWSM might include communications to a syslog server, Websense or N2H2 server, or AAA server.

The FWSM supports up to three equal cost routes on the same interface for load balancing.

Routes that identify a specific destination address take precedence over the default route.

To set a default route, enter the following command:

```
FWSM/contexta(config)# route gateway_interface 0 0 gateway_ip [metric]
```

The *metric* is the number of hops to *gateway_ip*. The default is 1 if you do not specify a metric.

For example, if the FWSM receives traffic that it does not have a route for, it sends the traffic out the outside interface to the router at 10.1.1.1:

```
FWSM/contexta(config)# route outside 0 0 10.1.1.1 1
```

Configuring Static Routes

Multiple context mode does not support dynamic routing, so you must use static routes for any networks to which the FWSM is not directly connected; for example, when there is a router between a network and the FWSM.

You might want to use static routes in single context mode in the following cases:

- Your networks use a different router discovery protocol from RIP or OSPF.
- Your network is small and you can easily manage static routes.
- You do not want the traffic or CPU overhead associated with routing protocols.

The simplest option is to configure a default route (see the previous section) to send all traffic to an upstream router, relying on the router to route the traffic for you. However, in some cases the default gateway might not be able to reach the destination network, so you must also configure more specific static routes. For example, if the default gateway is outside, then the default route cannot direct traffic to any inside networks that are not directly connected to the FWSM.

For transparent firewall mode, for traffic that originates on the FWSM and is destined for a non-directly connected network, you need to configure either a default route or static routes so the FWSM knows out of which interface to send traffic. Traffic that originates on the FWSM might include communications to a syslog server, Websense or N2H2 server, or AAA server. If you have servers that cannot all be reached through a single default route, then you must configure static routes.

The FWSM supports up to three equal cost routes on the same interface for load balancing.

Static routes take precedence over dynamic routes if they have the same metric (number of router hops).

To add a static route, enter the following command:

```
FWSM/contexta(config)# route gateway_interface dest_ip mask gateway_ip [metric]
```

The *metric* is the number of hops to *gateway_ip*. The default is 1 if you do not specify a metric.

The addresses you specify for the static route are the addresses that are in the packet before entering the FWSM and performing NAT.

For example, to send all traffic destined for 10.1.1.0/24 to the router (10.1.2.45) connected to the inside interface, enter the following command:

```
FWSM/contexta(config)# route inside 10.1.1.0 255.255.255.0 10.1.2.45 1
```

The following static routes are equal cost routes that direct traffic to three different routers on the outside interface. The FWSM sends 1/3 of the traffic to each router.

```
FWSM/contexta(config)# route outside 10.10.10.0 255.255.255.0 192.168.1.1
FWSM/contexta(config)# route outside 10.10.10.0 255.255.255.0 192.168.1.2
FWSM/contexta(config)# route outside 10.10.10.0 255.255.255.0 192.168.1.3
```

Configuring OSPF

Single context mode only

Routed firewall mode only

This section describes how to configure Open Shortest Path First (OSPF). For an overview of OSPF, see the following topic:

- OSPF Overview, page 8-4

To enable OSPF, see the following topic:

- Enabling OSPF, page 8-5

After you enable OSPF, you can configure advanced options according to the following topics:

- Redistributing Routes Between OSPF Processes, page 8-6
- Configuring OSPF Interface Parameters, page 8-9
- Configuring OSPF Area Parameters, page 8-11
- Configuring OSPF NSSA, page 8-12
- Configuring Route Summarization Between OSPF Areas, page 8-13
- Configuring Route Summarization When Redistributing Routes into OSPF, page 8-14
- Generating a Default Route, page 8-14
- Configuring Route Calculation Timers, page 8-15
- Logging Neighbors Going Up or Down, page 8-15
- Displaying OSPF Update Packet Pacing, page 8-16
- Monitoring OSPF, page 8-16
- Restarting the OSPF Process, page 8-17

OSPF Overview

Single context mode only

Routed firewall mode only

OSPF uses a link-state algorithm to build and calculate the shortest path to all known destinations. Each router in an OSPF area contains an identical link-state database, which is a list of each of the router usable interfaces and reachable neighbors.

The advantages of OSPF over RIP include the following:

- OSPF link-state database updates are sent less frequently than RIP updates, and the link-state database is updated instantly rather than gradually as stale information is timed out.
- Routing decisions are based on cost, which is an indication of the overhead required to send packets across a certain interface. The FWSM calculates the cost of an interface based on link bandwidth rather than the number of hops to the destination. The cost can be configured to specify preferred paths.

The disadvantage of shortest path first algorithms is that they require a lot of CPU cycles and memory.

The FWSM can run two processes of OSPF protocol simultaneously, on different sets of interfaces. You might want to run two processes if you have interfaces that use the same IP addresses (NAT allows these interfaces to coexist, but OSPF does not allow overlapping addresses). Or you might want to run one process on the inside, and another on the outside, and redistribute a subset of routes between the two processes. Similarly, you might need to segregate private addresses from public addresses.

Redistribution between the two OSPF processes is supported. Static and connected routes configured on OSPF-enabled interfaces on the FWSM can also be redistributed into the OSPF process. You cannot enable RIP on any of the same interfaces as the interfaces on which OSPF is enabled. Redistribution between RIP and OSPF is not supported.

The FWSM supports the following OSPF features:

- Support of intra-area, interarea, and external (type I and Type II) routes.
- Support of a virtual link.
- OSPF link-state advertisement (LSA) flooding.
- Authentication to OSPF packets (both password and MD5 authentication).
- Support for configuring the FWSM as a designated router or a designated backup router. The FWSM also can be set up as an area border router; however, the ability to configure the FWSM 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.

Enabling OSPF

Single context mode only

Routed firewall mode only

To enable OSPF, you need to create an OSPF routing process, specify the range of IP addresses associated with the routing process, then assign area IDs associated with that range of IP addresses.

To enable OSPF, follow these steps:

Step 1 To create an OSPF routing process, enter the following command:

```
FWSM(config)# router ospf process_id
```

This command enters the router configuration mode for this OSPF process.

The *process_id* is an internally used identifier for this routing process. It can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.

Step 2 To define the IP addresses on which OSPF runs and to define the area ID for that interface, enter the following command:

```
FWSM(config-router)# network ip_address mask area area_id
```

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
```

Redistributing Routes Between OSPF Processes

Single context mode only

Routed firewall mode only

The FWSM can control the redistribution of routes between OSPF routing processes. The FWSM matches and changes routes according to settings in the **redistribution** command or by using a route map. See also the "Generating a Default Route" section on page 8-14 for another use for route maps.

Note The FWSM cannot redistribute routes between routing protocols. However, the FWSM can redistribute static and connected routes.

This section includes the following topics:

- Adding a Route Map, page 8-6
- Redistributing Static, Connected, or OSPF Routes to an OSPF Process, page 8-8

Adding a Route Map

Single context mode only

Routed firewall mode only

To define a route map, follow these steps:

Step 1 To create a route map entry, enter the following command:

```
FWSM(config)# route-map name {permit | deny} [sequence_number]
```

Route map entries are read in order. You can identify the order using the *sequence_number* option, or the FWSM uses the order in which you add the entries.

Step 2 Enter one or more **match** commands:

- To match any routes that have a destination network that matches a standard ACL, enter the following command:

  ```
  FWSM(config-route-map)# match ip address acl_id [acl_id] [...]
  ```

  See the "Adding a Standard Access Control List" section on page 10-17 to add the standard ACL. If you specify more than one ACL, then the route can match any of the ACLs.

- To match any routes that have a specified metric, enter the following command:

  ```
  FWSM(config-route-map)# match metric metric_value
  ```

  The *metric_value* can be from 0 to 4294967295.

- To match any routes that have a next hop router address that matches a standard ACL, enter the following command:

  ```
  FWSM(config-route-map)# match ip next-hop acl_id [acl_id] [...]
  ```

  See the "Adding a Standard Access Control List" section on page 10-17 to add the standard ACL. If you specify more than one ACL, then the route can match any of the ACLs.

- To match any routes with the specified next hop interface, enter the following command:

  ```
  FWSM(config-route-map)# match interface vlan number [vlan number]
  ```

  If you specify more than one interface, then the route can match either interface.

- To match any routes that have been advertised by routers that match a standard ACL, enter the following command:

  ```
  FWSM(config-route-map)# match ip route-source acl_id [acl_id] [...]
  ```

  See the "Adding a Standard Access Control List" section on page 10-17 to add the standard ACL. If you specify more than one ACL, then the route can match any of the ACLs.

- To match the route type, enter the following command:

  ```
  FWSM(config-route-map)# match route-type {internal | external [type-1 | type-2]}
  ```

Step 3 Enter one or more **set** commands.

If a route matches the **match** commands, then the following **set** commands determine the action to perform on the route before redistributing it.

- To set the metric, enter the following command:

  ```
  FWSM(config-route-map)# set metric metric_value
  ```

  The *metric_value* can be a value between 0 and 294967295

- To set the metric type, enter the following command:

  ```
  FWSM(config-route-map)# set metric-type {type-1 | type-2}
  ```

- To set the next hop router IP address, enter the following command:

  ```
  FWSM(config-route-map)# set ip next-hop ip_address [ip_address] [...]
  ```

  The next hop must be an adjacent router. If you specify more than one address, if the interface associated with the first next hop address is down, then the next address is used.

The following example redistributes routes with a hop count equal to 1. The FWSM redistributes these routes as external link-state advertisements (LSAs) with a metric of 5, metric type of Type 1, and a tag equal to 1.

```
FWSM(config)# route-map 1-to-2 permit
FWSM(config-route-map)# match metric 1
FWSM(config-route-map)# set metric 5
FWSM(config-route-map)# set metric-type type-1
```

Redistributing Static, Connected, or OSPF Routes to an OSPF Process

Single context mode only

Routed firewall mode only

To redistribute static, connected, or OSPF routes from one process into another OSPF process, follow these steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to redistribute into by entering the following command:

```
FWSM(config)# router ospf process_id
```

Step 2 To specify the routes you want to redistribute, enter the following command:

```
FWSM(config-router)# redistribute {ospf process_id | static | connect}
[match {internal | external 1 | external 2}] [metric metric-value]
[metric-type {type-1 | type-2}] [tag tag_value] [subnets] [route-map map_name]
```

The **ospf** *process_id*, **static**, and **connect** keywords specify from where you want to redistribute routes.

You can either use the options in this command to match and set route properties, or you can use a route map. The **tag** and **subnets** options do not have equivalents in the **route-map** command. If you use both a route map and options in the **redistribute** command, then they must match.

The following example redistributes routes from OSPF process 1 into OSPF process 2 by matching routes with a metric equal to 1. The FWSM redistributes these routes as external link-state advertisements (LSAs) with a metric of 5, metric type of Type 1, and a tag equal to 1.

```
FWSM(config)# route-map 1-to-2 permit
FWSM(config-route-map)# match metric 1
FWSM(config-route-map)# set metric 5
FWSM(config-route-map)# set metric-type type-1
FWSM(config-route-map)# set tag 1
FWSM(config-route-map)# router ospf 2
FWSM(config-router)# redistribute ospf 1 route-map 1-to-2
```

The following example causes the specified OSPF process routes to be redistributed into OSPF process 109. The OSPF metric is remapped to 100.

```
FWSM(config)# router ospf 109
FWSM(config-router)# redistribute ospf 108 metric 100 subnets
FWSM(config-router)# redistribute static 108 metric 100 subnets
```

In the following example, the link-state cost is specified as 5, and the metric type is set to external, indicating that it has lower priority than internal metrics.

```
FWSM(config)# router ospf 1
FWSM(config-router)# redistribute ospf 2 metric 5 metric-type external
```

Configuring OSPF Interface Parameters

Single context mode only

Routed firewall mode only

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

To configure OSPF interface parameters, follow these steps:

Step 1 To enter the interface configuration mode, enter the following command:

```
FWSM(config)# interface interface_name
```

Step 2 Enter any of the following commands:

- To specify the authentication type for an interface, enter the following command:

  ```
  FWSM(config-interface)# ospf authentication [message-digest | null]
  ```

- To assign a password to be used by neighboring OSPF routers on a network segment that is using the OSPF simple password authentication, enter the following command:

  ```
  FWSM(config-interface)# ospf authentication-key key
  ```

  The *key* can be any continuous string of characters up to 8 bytes in length.

  The password created by this command is used as a key that is inserted directly into the OSPF header when the FWSM software originates routing protocol packets. A separate password can be assigned to each network on a per-interface basis. All neighboring routers on the same network must have the same password to be able to exchange OSPF information.

- To explicitly specify the cost of sending a packet on an OSPF interface, enter the following command:

  ```
  FWSM(config-interface)# ospf cost cost
  ```

  The *cost* is an integer from 1 to 65535.

- To set 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, enter the following command:

  ```
  FWSM(config-interface)# ospf dead-interval seconds
  ```

  The value must be the same for all nodes on the network.

- To specify the length of time between the hello packets that the FWSM sends on an OSPF interface, enter the following command:

  ```
  FWSM(config-interface)# ospf hello-interval seconds
  ```

  The value must be the same for all nodes on the network.

- To enable OSPF MD5 authentication, enter the following command:

  ```
  FWSM(config-interface)# ospf message-digest-key key_id md5 key
  ```

  Set the following values:

  - *key_id*—An identifier in the range from 1 to 255.
  - *key*—Alphanumeric password of up to 16 bytes.

  Usually, one key per interface is used to generate authentication information when sending packets and to authenticate incoming packets. The same key identifier on the neighbor router must have the same key value.

  We recommend that you not keep more than one key per interface. Every time you add a new key, you should remove the old key to prevent the local system from continuing to communicate with a hostile system that knows the old key. Removing the old key also reduces overhead during rollover.

- To set the priority to help determine the OSPF designated router for a network, enter the following command:

  ```
  FWSM(config-interface)# ospf priority number_value
  ```

  The *number_value* is between 0 to 255.

- To specify the number of seconds between link-state advertisement (LSA) retransmissions for adjacencies belonging to an OSPF interface, enter the following command:

  ```
  FWSM(config-interface)# ospf retransmit-interval seconds
  ```

  The *seconds* must be greater than the expected round-trip delay between any two routers on the attached network. The range is from 1 to 65535 seconds. The default is 5 seconds.

- To set the estimated number of seconds required to send a link-state update packet on an OSPF interface, enter the following command:

  ```
  FWSM(config-interface)# ospf transmit-delay seconds
  ```

  The *seconds* is from 1 to 65535 seconds. The default is 1 second.

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
```

View your configuration by entering the following command:

```
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

Single context mode only

Routed firewall mode only

You can configure several area parameters. These area parameters (shown in the following task table) include setting 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** command on the area border router to prevent it from sending summary link advertisement (LSAs type 3) into the stub area.

To specify area parameters for your network, follow these steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
FWSM(config)# router ospf process_id
```

Step 2 Enter any of the following commands:

- To enable authentication for an OSPF area, enter the following command:

  ```
  FWSM(config-router)# area area-id authentication
  ```

- To enable MD5 authentication for an OSPF area, enter the following command:

  ```
  FWSM(config-router)# area area-id authentication message-digest
  ```

- To define an area to be a stub area, enter the following command:

  ```
  FWSM(config-router)# area area-id stub [no-summary]
  ```

- To assign a specific cost to the default summary route used for the stub area, enter the following command:

  ```
  FWSM(config-router)# area area-id default-cost cost
  ```

  The *cost* is an integer from 1 to 65535. The default is 1.

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

Single context mode only

Routed firewall mode only

The OSPF implementation of a not-so-stubby-area (NSSA) is similar to an 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 an OSPF stub area because routes for the remote site could not be redistributed into the 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 for your network as needed to configure OSPF NSSA, follow these steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
FWSM(config)# router ospf process_id
```

Step 2 Enter any of the following commands:

- To define an NSSA area, enter the following command:

  ```
  FWSM(config-router)# area area-id nssa [no-redistribution]
  [default-information-originate]
  ```

- To summarize groups of addresses, enter the following command:

```
FWSM(config-router)# summary address ip_address mask [not advertise] [tag tag]
```

This command helps reduce the size of the routing table. Using this command for OSPF causes an OSPF autonomous system boundary Router (ASBR) to advertise one external route as an aggregate for all redistributed routes that are covered by the address.

OSPF does not support **summary-address 0.0.0.0 0.0.0.0**.

In the following example, the summary address 10.1.0.0 includes address 10.1.1.0, 10.1.2.0, 10.1.3.0, and so on. Only the address 10.1.0.0 is advertised in an external link-state advertisement:

```
FWSM(config-router)# summary-address 10.1.1.0 255.255.0.0
```

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

Single context mode only

Routed firewall mode only

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 advertises 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 define an address range for route summarization, follow these steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
FWSM(config)# router ospf process_id
```

Step 2 To set the address range, enter the following command:

```
FWSM(config-router)# area area-id range ip-address mask [advertise | not-advertise]
```

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

```
FWSM(config)# router ospf 1
FWSM(config-router)# area 17 range 12.1.0.0 255.255.0.0
```

Configuring Route Summarization When Redistributing Routes into OSPF

Single context mode only

Routed firewall mode only

When routes from other protocols are redistributed into OSPF, each route is advertised individually in an external LSA. However, you can configure the FWSM 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, follow these steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
FWSM(config)# router ospf process_id
```

Step 2 To set the summary address, enter the following command:

```
FWSM(config-router)# summary-address ip_address mask [not advertise] [tag tag]
```

OSPF does not support **summary-address 0.0.0.0 0.0.0.0**.

In the following example, the summary address 10.1.0.0 includes address 10.1.1.0, 10.1.2.0, 10.1.3.0, and so on. Only the address 10.1.0.0 is advertised in an external link-state advertisement:

```
FWSM(config)# router ospf 1
FWSM(config-router)# summary-address 10.1.0.0 255.255.0.0
```

Generating a Default Route

Single context mode only

Routed firewall mode only

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 generate a default route, follow these steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
FWSM(config)# router ospf process_id
```

Step 2 To force the autonomous system boundary router to generate a default route, enter the following command:

```
FWSM(config-router)# default-information originate [always] [metric metric-value]
[metric-type {1 | 2}] [route-map map-name]
```

This example shows how to generate a default route:

```
FWSM(config)# router ospf 2
FWSM(config-router)# default-information originate always
```

Configuring Route Calculation Timers

Single context mode only

Routed firewall mode only

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 route calculation timers, follow these steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
FWSM(config)# router ospf process_id
```

Step 2 To configure the route calculation time, enter the following command:

```
FWSM(config-router)# timers spf spf-delay spf-holdtime
```

The *spf-delay* is the delay time (in seconds) between when OSPF receives a topology change and when it starts an SPF calculation. It can be an integer from 0 to 65535. The default time is 5 seconds. A value of 0 means that there is no delay; that is, the SPF calculation is started immediately.

The *spf-holdtime* is the minimum time (in seconds) between two consecutive SPF calculations. It can be an integer from 0 to 65535. The default time is 10 seconds. A value of 0 means that there is no delay; that is, two SPF calculations can be done, one immediately after the other.

This example shows how to configure route calculation timers:

```
FWSM(config)# router ospf 1
FWSM(config-router)# timers spf 10 120
```

Logging Neighbors Going Up or Down

Single context mode only

Routed firewall mode only

By default, the system sends a system 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** 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.

To log neighbors going up or down, follow these steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
FWSM(config)# router ospf process_id
```

Step 2 To configure logging fir neighbors going up or down, enter the following command:

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

This example shows how to log neighbors:

```
FWSM(config)# router ospf 1
FWSM(config-router)# log-adj-changes detail
```

Displaying OSPF Update Packet Pacing

Single context mode only

Routed firewall mode only

OSPF update packets are automatically paced so they are not sent less than 33 milliseconds apart. Without pacing, some update packets could get lost in situations where the link is slow, a neighbor could not receive the updates quickly enough, or the router could run out of buffer space. For example, without pacing packets might be dropped if either of the following topologies exist:

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

Pacing is also used 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, enter the following command:

```
FWSM# show ip ospf flood-list vlan number
```

Monitoring OSPF

Single context mode only

Routed firewall mode only

You can display specific statistics such as the contents of IP routing tables, caches, and databases. You can use the information provided 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, as needed:

- To display general information about OSPF routing processes, enter the following command:

  ```
  FWSM# show ip ospf [process-id]
  ```

- To display the internal OSPF routing table entries to the area border router and autonomous system border router, enter the following command:

  ```
  FWSM# show ip ospf border-routers
  ```

- To display lists of information related to the OSPF database for a specific router, enter the following command:

  ```
  FWSM# show ip ospf [process-id [area-id]] database
  ```

- To display a list of LSAs waiting to be flooded over an interface (to observe OSPF packet pacing), enter the following command:

  ```
  FWSM# show ip ospf flood-list interface interface-type
  ```

- To display OSPF-related interface information, enter the following command:

  ```
  FWSM# show ip ospf interface [interface-type interface-number]
  ```

- To display OSPF neighbor information on a per-interface basis, enter the following command:

  ```
  FWSM# show ip ospf neighbor [interface-name] [neighbor-id] detail
  ```

- To display a list of all LSAs requested by a router, enter the following command:

  ```
  FWSM# show ip ospf request-list [neighbor] [interface] [interface-neighbor]
  ```

- To display a list of all LSAs waiting to be resent, enter the following command:

  ```
  FWSM# show ip ospf retransmission-list [neighbor] [interface] [interface-neighbor]
  ```

- To display a list of all summary address redistribution information configured under an OSPF process, enter the following command:

  ```
  FWSM# show ip ospf [process-id] summary-address
  ```

- To display OSPF-related virtual links information, enter the following command:

  ```
  FWSM# show ip ospf virtual-links
  ```

Restarting the OSPF Process

To restart an OSPF process, clear redistribution, or counters, enter the following command:

```
FWSM(config)# clear ip ospf pid {process | redistribution | counters
[neighbor [neighbor-interface] [neighbor-id]]}
```

Configuring RIP

Single context mode only

Routed firewall mode only

This section describes how to configure Route Information Protocol (RIP), and includes:

RIP Overview

Single context mode only

Routed firewall mode only

Devices that support RIP send routing-update messages at regular intervals and when the network topology changes. These RIP packets contain information about the networks that the devices can reach, as well as the number of routers or gateways that a packet must travel through to reach the destination address. RIP generates more traffic than OSPF, but is easier to configure initially.

RIP has advantages over static routes because the initial configuration is simple, and you do not need to update the configuration when the topology changes. The disadvantage to RIP is that there is more network and processing overhead than static routing.

The FWSM uses a limited version of RIP; it does not send out RIP updates that identify the networks that the FWSM can reach. However, you can enable one or both of the following methods:

- Passive RIP—The FWSM listens for RIP updates but does not send any updates about its networks out of the interface.

  Passive RIP allows the FWSM to learn about networks to which it is not directly connected.

- Default Route Updates—Instead of sending normal RIP updates that describe all the networks reachable through the FWSM, the FWSM sends a default route to participating devices that identifies the FWSM as the default gateway.

  You can use the default route option with passive RIP, or alone. You might use the default route option alone if you use static routes on the FWSM, but do not want to configure static routes on downstream routers. Typically, you would not enable the default route option on the outside interface, because the FWSM is not typically the default gateway for the upstream router.

Enabling RIP

Single context mode only

Routed firewall mode only

To enable RIP on an interface, enter the following command:

```
FWSM(config)# rip interface_name {default | passive} [version {1 | 2
[authentication {text | md5} key key_id]}]
```

You can both types of RIP on an interface by entering the command two times, one for each method.

If you want to use both modes, then enter the **rip** command two times with different modes for a given interface. For example, enter the following commands:

```
FWSM(config)# rip inside default version 2 authentication md5 scorpius 1
FWSM(config)# rip inside passive version 2 authentication md5 scorpius 1
```

If you want to enable passive RIP on all interfaces, but only enable default routes on the inside interface, enter the following commands:

```
FWSM(config)# rip inside default version 2 authentication md5 scorpius 1
FWSM(config)# rip inside passive version 2 authentication md5 scorpius 1
FWSM(config)# rip outside passive version 2 authentication md5 scorpius 1
```

Note Before testing your configuration, flush the ARP caches on any routers connected to the FWSM. For Cisco routers, use the **clear arp** command to flush the ARP cache.

Configuring the DHCP Server

This section describes how to use the Dynamic Host Configuration Protocol (DHCP) server provided by the FWSM. It includes the following topics:

Enabling the DHCP Server

The FWSM can act as a DHCP server. DHCP is a protocol that supplies network settings to hosts including the host IP address, the default gateway, and a DNS server.

Note The FWSM DHCP server does not support BOOTP requests.

Note For multiple context mode, you cannot enable the DHCP server or DHCP relay on an interface that is used by more than one context (a shared VLAN).

The DHCP server can be enabled on any interface. Clients must be directly connected to the FWSM and cannot send requests through another relay agent or a router.

To enable the DHCP server on a given FWSM interface, follow these steps:

Step 1 To create a DHCP address pool, enter the following command:

```
FWSM/contexta(config)# dhcpd address ip_address-ip_address interface_name
```

The FWSM assigns a client one of the addresses from this pool to use for a given length of time. These addresses are the local untranslated addresses for the directly connected network.

The address pool must be on the same subnet as the FWSM interface.

Step 2 (Optional) To specify the IP address(es) of the DNS server(s) the client will use, enter the following command:

```
FWSM/contexta(config)# dhcpd dns dns1 [dns2]
```

You can specify up to two DNS servers.

Step 3 (Optional) To specify the IP address(es) of the WINS server(s) the client will use, enter the following command:

```
FWSM/contexta(config)# dhcpd wins wins1 [wins2]
```

You can specify up to two WINS servers.

Step 4 (Optional) To change the lease length to be granted to the client, enter the following command:

```
FWSM/contexta(config)# dhcpd lease lease_length
```

This lease equals the amount of time (in seconds) the client can use its allocated IP address before the lease expires. Enter a value between 0 to 1,048,575. The default value is 3600 seconds.

Step 5 (Optional) To configure the domain name the client uses, enter the following command:

```
FWSM/contexta(config)# dhcpd domain domain_name
```

Step 6 To enable the DHCP daemon within the FWSM to listen for DHCP client requests on the enabled interface, enter the following command:

```
FWSM/contexta(config)# dhcpd enable interface_name
```

For example, to assign the range 10.0.1.101 to 10.0.1.110 to hosts connected to the inside interface, enter the following commands:

```
FWSM/contexta(config)# dhcpd address 10.0.1.101-10.0.1.110 inside
FWSM/contexta(config)# dhcpd dns 209.165.201.2 209.165.202.129
FWSM/contexta(config)# dhcpd wins 209.165.201.5
FWSM/contexta(config)# dhcpd lease 3000
FWSM/contexta(config)# dhcpd domain example.com
FWSM/contexta(config)# dhcpd enable inside
```

Using Cisco IP Phones with a DHCP Server

Enterprises with small branch offices that implement a Cisco IP Telephony Voice over IP (VoIP) solution typically implement Cisco CallManager at a central office to control Cisco IP Phones at small branch offices. This implementation allows centralized call processing, reduces the equipment required, and eliminates the administration of additional Cisco CallManager and other servers at branch offices.

Cisco IP Phones download their configuration from a TFTP server. When a Cisco IP Phone starts, if it does not have both the IP address and TFTP server IP address preconfigured, it sends a request with option 150 or 66 to the DHCP server to obtain this information.

- DHCP option 150 provides the IP addresses of a list of TFTP servers
- DHCP option 66 gives the IP address or the host name of a single TFTP server.

Cisco IP Phones might also include DHCP option 3 in their requests, which lists the IP addresses of default routers.

Cisco IP Phones might include both option 150 and 66 in a single request. In this case, the FWSM DHCP server provides values for both options in the response if they are configured on the FWSM.

You can configure the FWSM to send information for most options listed in RFC 2132. The following table shows the syntax for any option number, as well as the syntax for commonly-used options 66,150, and 3:

- To provide information for DHCP requests that include an option number as specified in RFC-2132, enter the following command:

  ```
  FWSM/contexta(config)# dhcpd option number string
  ```

- To provide the IP address or name of a TFTP server for option 66, enter the following command:

  ```
  FWSM/contexta(config)# dhcpd option 66 ascii server_name
  ```

- To provide the IP address or names of one or two TFTP servers for option 150, enter the following command:

  ```
  FWSM/contexta(config)# dhcpd option 150 ip server_ip1 [server_ip2]
  ```

  The *server_ip1* is the IP address or name of the primary TFTP server while *server_ip2* is the IP address or name of the secondary TFTP server. A maximum of two TFTP servers can be identified using option 150.

- To provide a list of router IP addresses for option 3, enter the following command:

  ```
  FWSM/contexta(config)# dhcpd option 3 ip router_ip1 [router_ip2] [...]
  ```

Configuring DHCP Relay

Routed firewall mode only

A DHCP relay agent allows the FWSM to forward DHCP requests from clients to a router connected to a different interface.

The following restrictions apply to the use of the DHCP relay agent:

- The relay agent cannot be enabled if the DHCP server feature is also enabled.
- Clients must be directly connected to the FWSM and cannot send requests through another relay agent or a router.
- For multiple context mode, you cannot enable DHCP relay on an interface that is used by more than one context (a shared VLAN).

To enable DHCP relay, follow these steps:

Step 1 To set the IP address of a DHCP server on a different interface from the DHCP client, enter the following command:

```
FWSM/contexta(config)# dhcprelay server ip_address
```

You can use this command up to 10 times to identify up to 10 servers.

Step 2 To enable DHCP relay on the interface connected to the clients, enter the following command:

```
FWSM/contexta(config)# dhcprelay enable interface
```

Step 3 (Optional) To set the number of seconds allowed for relay address negotiation, enter the following command:

```
FWSM/contexta(config)# dhcprelay timeout seconds
```

Step 4 (Optional) To change the first default router address in the packet sent from the DHCP server to the address of the FWSM interface, enter the following command:

```
FWSM/contexta(config)# dhcprelay setroute interface_name
```

This action allows the client to set its default route to point to the FWSM even if the DHCP server specifies a different router.

If there is no default router option in the packet, the FWSM adds one containing the interface address.

The following example enables the FWSM to forward DHCP requests from clients connected to the inside interface to a DHCP server on the outside interface:

```
FWSM/contexta(config)# dhcprelay server 201.168.200.4
FWSM/contexta(config)# dhcprelay enable inside
FWSM/contexta(config)# dhcprelay setroute inside
```

Configuring Network Address Translation

Routed firewall mode only

This chapter describes Network Address Translation (NAT). In routed firewall mode, the Firewall Services Module (FWSM) typically performs NAT between each network.

Note In transparent firewall mode, both the inside and outside network are the same network, and the FWSM does not perform NAT. See the "Configuring Connection Limits for Non-NAT Configurations" section on page 6-9 for connection limits for which you must configure a NAT statement in transparent firewall mode.

This chapter contains the following sections:

NAT Overview

This section describes how NAT works on the FWSM, and includes the following topics:

Introduction to NAT

Address translation substitutes the local address in a packet with a global address that is routable on the destination network. In this document, all types of translation are generally referred to as "NAT."

On the FWSM, you must specifically configure some interfaces to either use or to bypass NAT. For example, when hosts on a higher security interface (inside) access hosts on a lower security interface (outside), you must configure NAT on the inside hosts *or* specifically configure the inside hosts to bypass NAT (See the "Configuring Interfaces" section on page 6-6 for more information about security levels).

Note When discussing NAT, the terms *inside* and *outside* are relative, and represent the security relationship between any two interfaces. The higher security level is inside and the lower security level is outside; for example, interface 1 is at 60 and interface 2 is at 50, so interface 1 is "inside" and interface 2 is "outside."

An inside host can communicate with the untranslated local address of the outside host without any special configuration on the outside interface. However, you can also optionally configure NAT on the outside network.

Interfaces that are on the same security level that you have allowed to communicate do not have to perform NAT. You can, however, optionally configure NAT for these interfaces. (See the "Allowing Communication Between Interfaces on the Same Security Level" section on page 6-8 for more information.) In this case, there is no inside or outside when performing NAT between two interfaces.

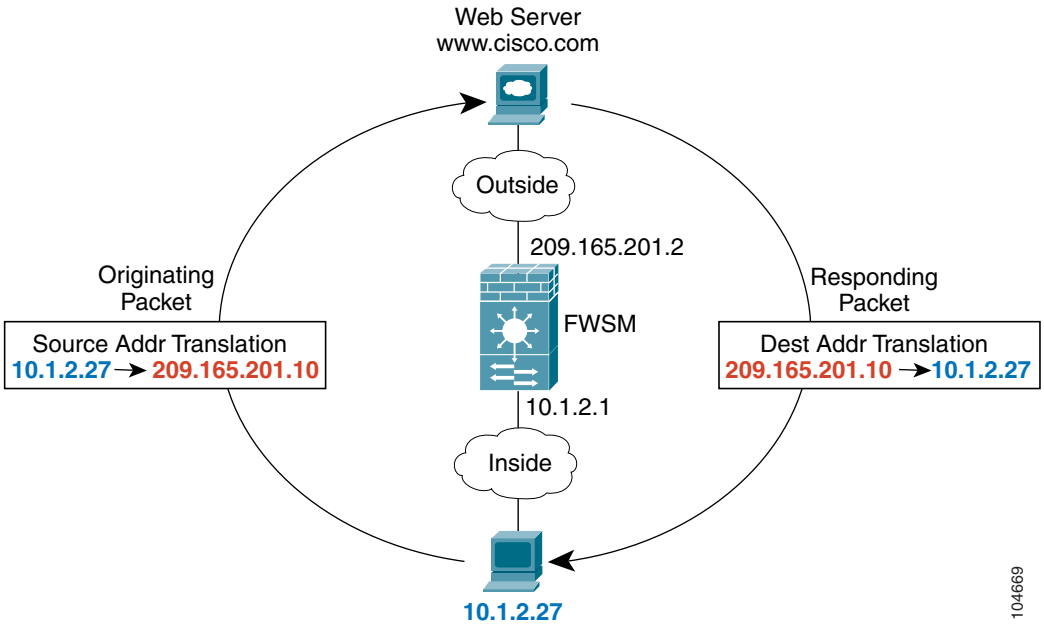
Some of benefits of NAT are as follows:

- You can use private addresses on your inside networks. Private addresses are not routable on the Internet. (See the "Private Networks" section on page D-2 for more information.)
- NAT hides the local addresses from other networks, so attackers cannot learn the real address of a host.
- You can resolve IP routing problems such as overlapping addresses.

Note See Table 13-1 on page 13-2 for information about protocols that are not supported by NAT.

Figure 9-1 shows a typical NAT scenario, with a private network on the inside. When the inside host sends a packet to a web server, the local source address of the packet is changed to a routable global address. When the server responds, it sends the response to the global address, and the FWSM receives the packet. The FWSM then translates the global address to the local address before sending it on to the host.

Figure 9-1 NAT Example



See the following commands for this example:

```
FWSM/contexta(config)# nat (inside) 1 10.1.2.0 255.255.255.0
FWSM/contexta(config)# global (outside) 1 209.165.201.1-209.165.201.15
```

NAT Types

You can implement address translation as dynamic NAT, Port Address Translation (PAT), static NAT, or static PAT or as a mix of these types. You can also bypass NAT. See the following sections for information about each type:

- Dynamic NAT, page 9-3
- PAT, page 9-4
- Static NAT, page 9-5
- Static PAT, page 9-5
- Bypassing NAT, page 9-7

Dynamic NAT

Dynamic NAT translates a group of local addresses to a pool of global addresses that are routable on the destination network. The global pool can include fewer addresses than the local group. When a local host accesses the destination network, the FWSM assigns it an IP address from the global pool. Because the

translation is only in place for the duration of the connection, a given user does not keep the same IP address after the translation times out (see the **timeout xlate** command in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*). Users on the destination network, therefore, cannot reliably initiate a connection to a host that uses dynamic NAT (even if the connection is allowed by an access control list (ACL)). Not only can you not predict the global IP address of the host, but the FWSM does not create a translation at all unless the local host is the initiator. See "Static NAT" or "Static PAT" below for reliable access to hosts.

Note For the duration of the translation, a global host can initiate a connection to the local host if an ACL allows it. Because the address is unpredictable, a connection to the host is unlikely. However in this case, you can rely on the security of the ACL.

Dynamic NAT has these disadvantages:

- If the global pool has fewer addresses than the local group, you could run out of addresses if the amount of traffic is more than expected.

  Use PAT if this event occurs often, because PAT provides over 64,000 translations using ports of a single address.

- You have to use a large number of routable addresses in the global pool; if the destination network requires registered addresses, such as the Internet, you might encounter a shortage of usable addresses.

The advantage of dynamic NAT is that some protocols cannot use PAT. PAT does not work with some applications that have a data stream on one port and the control path on another, such as some multimedia applications. See the "Inspection Engine Overview" section on page 13-1 for more information about NAT and PAT support.

PAT

PAT translates multiple local addresses to a single global IP address. Specifically, the FWSM translates the local address and local port for multiple connections and/or hosts to a single global address and a unique port (above 1024). When a local host connects to the destination network on a given source port, the FWSM assigns the global IP address to it and a unique port number. Each host receives the same IP address, but because the source port numbers are unique, the responding traffic, which includes the IP address and port number as the destination, can be sent to the correct host. Because there are over 64,000 ports available, you are unlikely to run out of addresses, which can happen with dynamic NAT. Because the translation is specific to the local address and local port, each connection, which generates a new source port, requires a separate translation. For example, 10.1.1.1:1025 requires a separate translation from 10.1.1.1:1026.

The translation is only in place for the duration of the connection, so a given user does not keep the same global IP address port number after the translation times out (see the **timeout xlate** command in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*). Users on the destination network, therefore, cannot reliably initiate a connection to a host that uses PAT (even if the connection is allowed by an ACL). Not only can you not predict the local or global port number of the host, but the FWSM does not create a translation at all unless the local host is the initiator. See "Static NAT" or "Static PAT" below for reliable access to hosts.

PAT lets you use a single global address, thus conserving routable addresses. You can even use the FWSM interface IP address as the PAT address. PAT does not work with some multimedia applications that have a data stream that is different from the control path. See the "Inspection Engine Overview" section on page 13-1 for more information about NAT and PAT support.

Note For the duration of the translation, a global host can initiate a connection to the local host if an ACL allows it. Because the port address (both local and global) is unpredictable, a connection to the host is unlikely. However in this case, you can rely on the security of the ACL.

Static NAT

Static NAT translates each local address to a fixed global address. With dynamic NAT and PAT, each host uses a different address or port after the translation times out. Because the global address is the same for each consecutive connection with static NAT, and a persistent translation rule exists, static NAT allows hosts on the global network to initiate traffic to a local host (if there is an ACL that allows it).

The main difference between dynamic NAT and a range of addresses for static NAT is that static NAT allows a global host to initiate a connection to a local host (if there is an ACL that allows it), while dynamic NAT does not. You also need an equal number of global addresses as local addresses with static NAT.

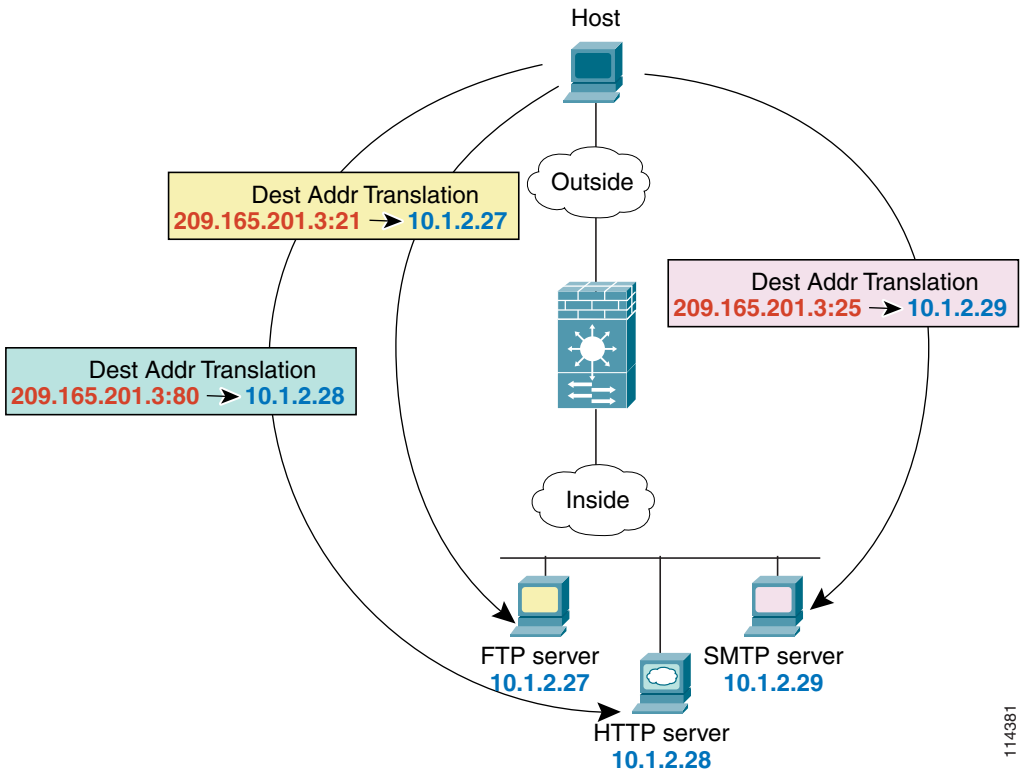
Static PAT

Static PAT is the same as static NAT, except it lets you specify the protocol (TCP or UDP) and port for the local and global addresses.

This feature lets you identify the same global address across many different static statements, so long as the port is different for each statement (you cannot use the same global address for multiple static *NAT* statements).

For example, if you want to provide a single address for global users to access FTP, HTTP, and SMTP, but these are all actually different servers on the local network, you can specify static PAT statements for each server that uses the same global IP address, but different ports (see Figure 9-2).

Figure 9-2 Static PAT



See the following commands for this example:

```
FWSM/contexta(config)# static (inside,outside) tcp 209.165.201.3 ftp 10.1.2.27 ftp netmask
255.255.255.255
FWSM/contexta(config)# static (inside,outside) tcp 209.165.201.3 http 10.1.2.28 http
netmask 255.255.255.255
FWSM/contexta(config)# static (inside,outside) tcp 209.165.201.3 smtp 10.1.2.29 smtp
netmask 255.255.255.255
```

If the application used by the server requires an inspection engine to allow data channels on other ports, such as FTP, then the server needs translation for other ports. Other protocols that require inspection engines for data channels include TFTP, RTSP, and Skinny. See Chapter 13, "Configuring Application Protocol Inspection," for a complete list of protocols that require inspection engines. For example, add the following line to the above configuration to translate all other ports from the FTP server at 10.1.2.27:

```
FWSM/contexta(config)# nat (inside) 1 10.1.2.27 255.255.255.255
FWSM/contexta(config)# global (outside) 1 209.165.201.3
```

The above configuration also allows the FTP server to initiate connections, if desired.

You can also use static PAT to translate a well-known port to a non-standard port or vice versa. For example, if your inside web servers use port 8080, you can allow outside users to connect to port 80, and then translate them to the 8080 port. Similarly, if you want to provide extra security, you can tell your web users to connect to non-standard port 6785, and then translate them to port 80 on the local network.

Bypassing NAT

When hosts on a higher security interface (inside) access hosts on a lower security interface (outside), you must configure NAT on the inside hosts or specifically configure the inside interface to bypass NAT. You might want to bypass NAT in the following circumstances:

- You do not want the complication of NAT.
- You are using an application that does not support NAT (see the "Inspection Engine Overview" section on page 13-1 for information about inspection engines that do not support NAT).
- You are using a transparent firewall and want to set connection limits.
- You are using same security interfaces and want to set connection limits.

You can configure an interface to bypass NAT using three methods. All methods achieve compatibility with inspection engines and simplification of your addressing. However, each method offers slightly different capabilities, as follows:

- Identity NAT—When you configure identity NAT (which is similar to dynamic NAT), you do not specify global addresses, and therefore you do not specify a single global interface; you must use identity NAT for connections through all interfaces. Therefore, you cannot choose to perform normal translation on local addresses when you access interface A, but use identity NAT when accessing interface B. Regular dynamic NAT, on the other hand, lets you specify a particular global interface on which to translate the addresses. Make sure that the local addresses for which you use identity NAT are routable on all networks that are available according to your ACLs.

  For identity NAT, even though the translated address is the same as the local address, you cannot initiate a connection from the outside to the inside (even if the interface ACL allows it). Use static identity NAT or NAT exemption for this functionality. For same security interfaces, however, you can initiate connections both ways.

- Static identity NAT—Static identity NAT lets you specify the global interface on which you want to allow the local addresses to appear, so you can use identity NAT when you access interface A, and use regular translation when you access interface B. Static identity NAT also lets you use policy NAT, which identifies the local and destination addresses when determining the local traffic to translate (see the "Policy NAT" section on page 9-8 for more information about policy NAT). For example, you can use static identity NAT for an inside address when it accesses the outside interface and the destination is server A, but use a normal translation when accessing the outside server B.

- NAT exemption— NAT exemption allows both local and global hosts to initiate connections. Like identity NAT, you do not specify global addresses, and therefore you do not specify a single global interface; you must use NAT exemption for connections through all interfaces. However, NAT exemption does allow you to specify the local and destination addresses when determining the local traffic to translate (similar to policy NAT), so you have greater control using NAT exemption. However unlike policy NAT, NAT exemption does not consider the ports in the ACL.

Note In multiple context mode, you cannot initiate connections from an interface shared between contexts when you use NAT exemption for the destination address. The classifier can only assign packets from a shared interface to a context when you configure a static statement for the destination address. For example, if you share the outside interface, you cannot use NAT exemption on an inside interface if you want outside traffic to reach the inside addresses. The classifier only looks at static statements where the global interface matches the source interface of the packet. Because NAT exemption does not identify a global interface, the classifier does not consider those NAT statements for classification purposes.

Policy NAT

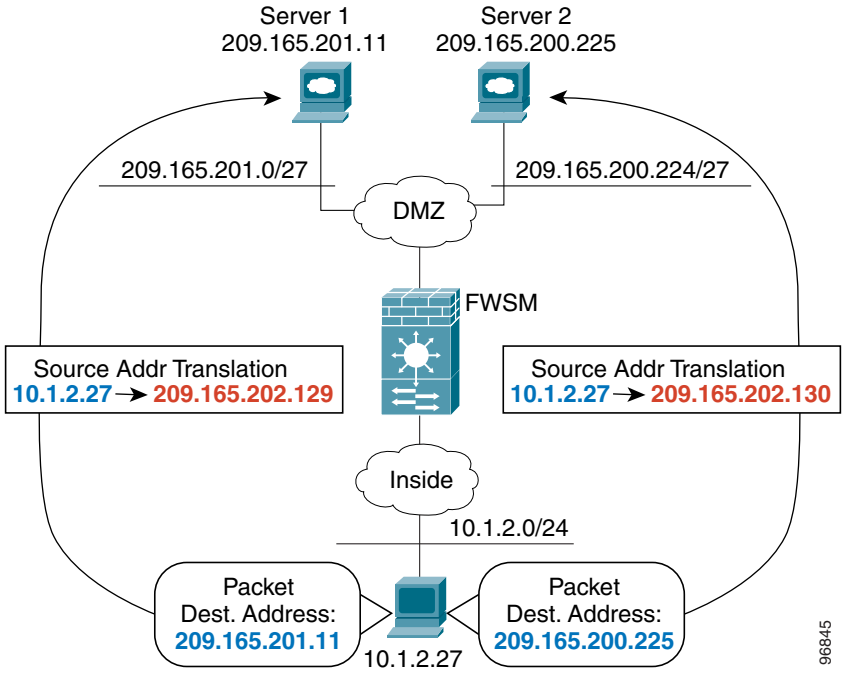
Policy NAT lets you identify local traffic for address translation by specifying the source and destination addresses in an extended ACL. You can also optionally specify the source and destination ports. Regular NAT can only consider the local addresses.

Note All types of NAT support policy NAT except for NAT exemption. NAT exemption uses an ACL to identify the local addresses, but differs from policy NAT in that the ports are not considered. See the "Bypassing NAT" section on page 9-29 for other differences.

With policy NAT, you can create multiple NAT or static statements that identify the same local address as long as the source/port and destination/port combination is unique for each statement. You can then match different global addresses to each source/port and destination/port pair.

Figure 9-3 shows a host on the 10.1.2.0/24 network accessing two different servers. When the host accesses the server at 209.165.201.11, the local address is translated to 209.165.202.129. When the host accesses the server at 209.165.200.225, the local address is translated to 209.165.202.130 so that the host appears to be on the same network as the servers, which can help with routing.

Figure 9-3 Policy NAT with Different Destination Addresses

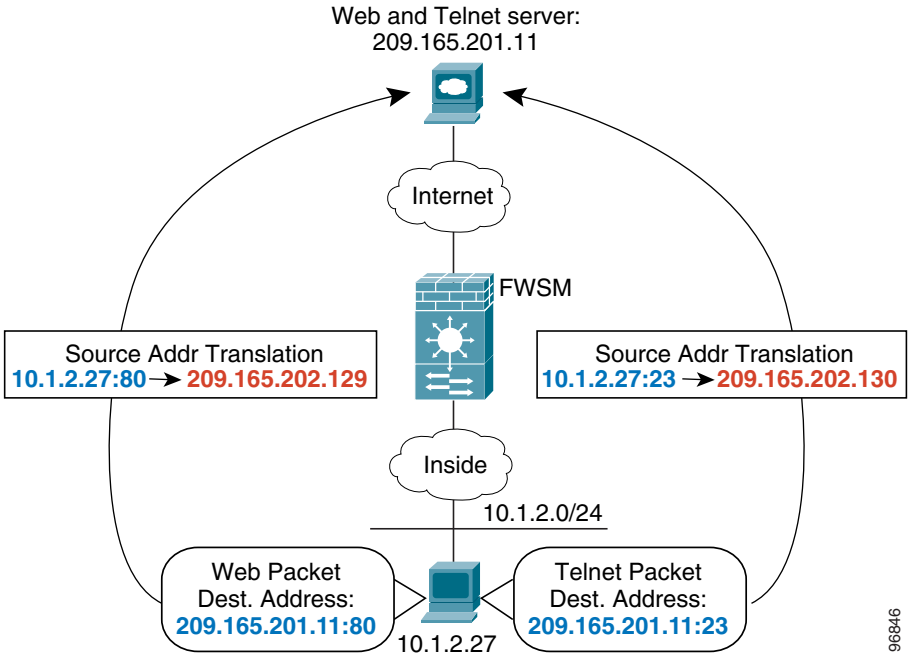


See the following commands for this example:

```
FWSM/contexta(config)# access-list NET1 permit ip 10.1.2.0 255.255.255.0 209.165.201.0
255.255.255.224
FWSM/contexta(config)# access-list NET2 permit ip 10.1.2.0 255.255.255.0 209.165.200.224
255.255.255.224
FWSM/contexta(config)# nat (inside) 1 access-list NET1
FWSM/contexta(config)# global (outside) 1 209.165.202.129
FWSM/contexta(config)# nat (inside) 2 access-list NET2
FWSM/contexta(config)# global (outside) 2 209.165.202.130
```

Figure 9-4 shows the use of source and destination ports. The host on the 10.1.2.0/24 network accesses a single host for both web services and Telnet services. When the host accesses the server for web services, the local address is translated to 209.165.202.129. When the host accesses the same server for Telnet services, the local address is translated to 209.165.202.130.

Figure 9-4 Policy NAT with Different Destination Ports



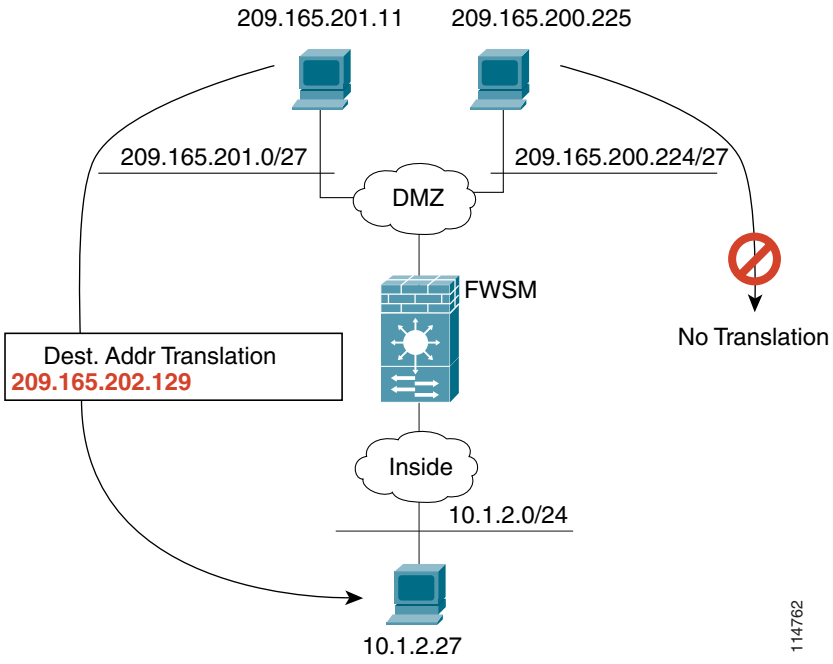
See the following commands for this example:

```
FWSM/contexta(config)# access-list WEB permit tcp 10.1.2.0 255.255.255.0 209.165.201.11
255.255.255.255 eq 80
FWSM/contexta(config)# access-list TELNET permit tcp 10.1.2.0 255.255.255.0 209.165.201.11
255.255.255.255 eq 23
FWSM/contexta(config)# nat (inside) 1 access-list WEB
FWSM/contexta(config)# global (outside) 1 209.165.202.129
FWSM/contexta(config)# nat (inside) 2 access-list TELNET
FWSM/contexta(config)# global (outside) 2 209.165.202.130
```

For policy static NAT (and for NAT exemption, which also uses an ACL to identify traffic), both local and global hosts can originate traffic. For locally originated traffic, the NAT ACL specifies the local addresses and the *destination* addresses, but for globally originated traffic, the ACL identifies the local addresses and the *source* addresses of global hosts who are allowed to connect to the local host using this translation. Figure 9-5 shows a global host connecting to a local host. The local host has a policy

static NAT translation that translates the local address only for traffic to and from the 209.165.201.0/27 network. A translation does not exist for the 209.165.200.224/27 network, so the local host cannot connect to that network, nor can a host on that network connect to the local host.

Figure 9-5 Policy Static NAT with Destination Address Translation



Note Policy NAT does not support SQL*Net, but it is supported by regular NAT. See the "Inspection Engine Overview" section on page 13-1 for information about NAT support for other protocols.

Note The number of access control entries (ACEs) used in policy NAT statements is limited. See the "Maximum Number of ACEs" section on page 10-7 for information about limits on certain types of rules.

Outside NAT

When hosts on a lower security interface (outside) access hosts on a higher security interface (inside), you do not have to perform NAT on the outside hosts. (See the "Configuring Interfaces" section on page 6-6 for more information about security levels.) You can, however, optionally configure NAT on outside interfaces so that the outside host address is translated. Because the inside host is also typically translated using a static NAT statement, both host addresses are translated.

If you configure dynamic NAT or PAT (**nat** and **global** commands) for any hosts on an outside interface when they access hosts on a given inside interface, then for any traffic between those two interfaces, the NAT requirements change for the outside interface. Namely, the outside interface takes on the NAT requirements of an inside interface, as follows:

- No traffic can originate on the outside interface without being translated (or being configured to bypass NAT).

  This requirement is true even if the dynamic NAT statement includes only a few addresses. Other addresses not included in the dynamic NAT statement require a NAT configuration to originate connections, even if the NAT configuration is to bypass NAT and use the original addresses.

- No traffic from the specified inside interface can access hosts behind the outside interface unless you configure a static NAT statement, static identity NAT statement, or a NAT exemption statement for the outside hosts.

If you only configure static NAT for the outside interface, these restrictions do not apply. Traffic between the outside interface and a different inside interface is not affected.

Connection limitations that are set using NAT commands are not applied for outside NAT. (See the "Setting Connection Limits in the NAT Configuration" section on page 9-16.)

You might want to use outside NAT, for example, to accommodate overlapping addresses. (See the "Overlapping Networks" section on page 9-33.)

NAT and Same Security Level Interfaces

NAT is not required between same security level interfaces (see the "Allowing Communication Between Interfaces on the Same Security Level" section on page 6-8 to enable same security communication). However, you can optionally configure NAT if desired. Because there is no "inside" and "outside" when configuring NAT between two interfaces at the same security, connection limits that you set in the NAT configuration apply in both directions.

If you configure dynamic NAT or PAT (**nat** and **global** commands) for any hosts on a local interface when they access hosts on a given same security interface, then for any traffic between those two interfaces, the NAT requirements change for the local interface. Namely, the local interface takes on the NAT requirements of an inside interface, as follows:

- No traffic can originate on the local interface without being translated (or being configured to bypass NAT).

  This requirement is true even if the dynamic NAT statement includes only a few addresses. Other addresses not included in the dynamic NAT statement require a NAT configuration to originate connections, even if the NAT configuration is to bypass NAT and use the original addresses.

- No traffic from the specified same security interface can access hosts behind the local interface unless you configure a static NAT statement, a NAT exemption statement, or an identity NAT statement for the local hosts.

If you only configure static NAT, identity NAT, or NAT exemption for the local interface, these restrictions do not apply. Traffic between the local interface and a different same security interface is not affected.

You might want to configure NAT exemption or identity NAT on same security interfaces to set connection limits. (See the "Setting Connection Limits in the NAT Configuration" section on page 9-16.)

Note The FWSM does not support VoIP inspection engines when you configure NAT on same security interfaces. These inspection engines include Skinny, SIP, and H.323. See the "Inspection Support" section on page 13-2 for supported inspection engines.

Order of NAT Commands Used to Match Local Addresses

The FWSM matches local traffic to NAT commands in the following order:

1. NAT exemption (**nat 0 access-list**)—In order, until the first match. Identity NAT is not included in this category; it is included in the regular static NAT or regular NAT category. We do not recommend overlapping addresses in NAT exemption statements because unexpected results can occur.
2. Static NAT and Static PAT (regular and policy) (**static**)—In order, until the first match. Static identity NAT is included in this category. We do not recommend overlapping addresses in static statements because unexpected results can occur.
3. Policy dynamic NAT (**nat access-list**)—In order, until the first match. Overlapping addresses are allowed.
4. Regular dynamic NAT (**nat**)—Best match. Regular identity NAT is included in this category. The order of the NAT commands does not matter; the NAT statement that best matches the local traffic is used. For example, you can create a general statement to translate all addresses (0.0.0.0) on an interface. If you want to translate a subset of your network (10.1.1.1) to a different address, then you can create a statement to translate only 10.1.1.1. When 10.1.1.1 makes a connection, the specific statement for 10.1.1.1 is used because it matches the local traffic best. We do not recommend using overlapping statements; they use more memory and can slow the performance of the FWSM.

Maximum Number of NAT Statements

The FWSM supports the following numbers of **nat**, **global**, and **static** commands divided between all contexts or in single mode:

- **nat** command—2 K
- **global** command—1,051
- **static** command—2 K

The FWSM also supports up to 3942 access control entries (ACEs) in ACLs used for policy NAT for single mode, and 7,272 ACEs for multiple mode.

Global Address Guidelines

When you translate the local address to a global address, you can use the following global addresses:

- Addresses on the same network as the global interface.

  If you use addresses on the same network as the global interface (through which traffic exits the FWSM), the FWSM uses proxy ARP to answer any requests for translated addresses, and thus intercepts traffic destined for a local address. This solution simplifies routing, because the FWSM does not have to be the gateway for any additional networks. However, this approach does put a limit on the number of available addresses used for translations.

  For PAT, you can even use the IP address of the global interface.

- Addresses on a unique network.

If you need more addresses than are available on the global interface network, you can identify addresses on a different subnet. The FWSM uses proxy ARP to answer any requests for translated addresses, and thus intercepts traffic destined for a local address. If you use OSPF, and you advertise routes on the global interface, then the FWSM advertises the translated addresses. If the global interface is passive (not advertising routes) or you are using static routing, then you need to add a static route on the upstream router that sends traffic destined for the translated addresses to the FWSM.

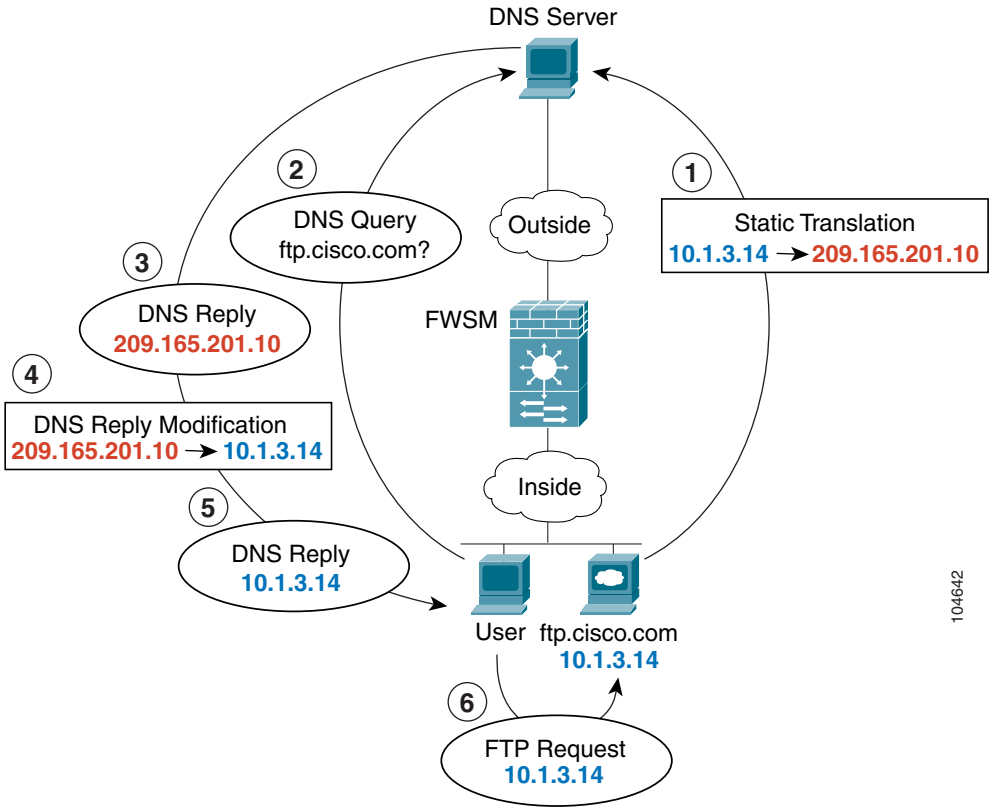
DNS and NAT

You might need to configure the FWSM to modify DNS replies by replacing the address in the reply with an address that matches the NAT configuration. You can configure DNS modification when you configure each NAT translation.

For example, a DNS server is accessible from the outside interface. A server, ftp.cisco.com, is on the inside interface. You configure the FWSM to statically translate the ftp.cisco.com local address (10.1.3.14) to a global address (209.165.201.10) that is visible on the outside network (See Figure 9-6). In this case, you want to enable DNS reply modification on this static statement so that inside users who have access to ftp.cisco.com using the local address receive the local address from the DNS server, and not the global address.

When an inside host sends a DNS request for the address of ftp.cisco.com, the DNS server replies with the global address (209.165.201.10). The FWSM refers to the static statement for the inside server and translates the address inside the DNS reply to 10.1.3.14. If you do not enable DNS reply modification, then the inside host attempts to send traffic to 209.165.201.10 instead of accessing ftp.cisco.com directly.

Figure 9-6 DNS Reply Modification

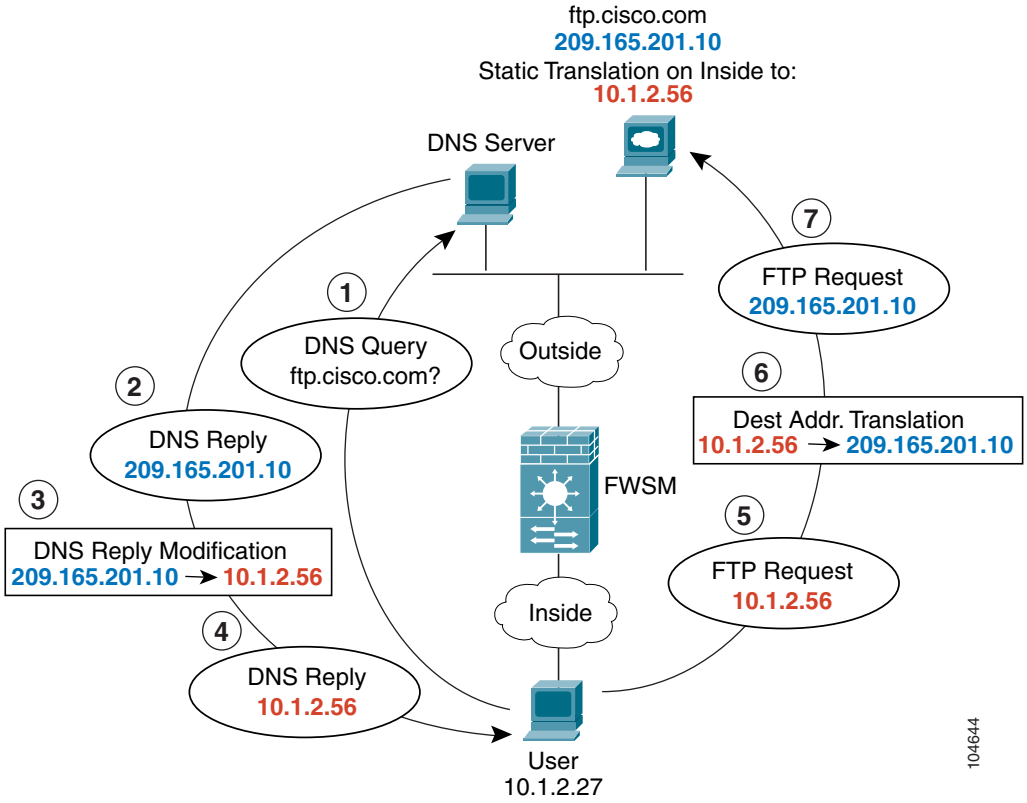


See the following command for this example:

```
FWSM/contexta(config)# static (inside,outside) 209.165.201.10 10.1.3.14 netmask
255.255.255.255 dns
```

Figure 9-6 shows a web server and DNS server on the outside. The FWSM has a static translation for the outside server. In this case, when an inside user requests the address for ftp.cisco.com from the DNS server, the DNS server responds with the real local address, 209.165.20.10. Because you want inside users to use the translated global address for ftp.cisco.com (10.1.2.56) you need to configure DNS reply modification for the static translation.

Figure 9-7 DNS Reply Modification Using Outside NAT



See the following command for this example:

```
FWSM/contexta(config)# static (outside,inside) 10.1.2.56 209.165.201.10 netmask
255.255.255.255 dns
```

Setting Connection Limits in the NAT Configuration

The NAT configuration lets you set some options for traffic that cannot be set anywhere else, including the following:

- Setting the maximum connections—The maximum number of simultaneous TCP and/or UDP connections for the entire subnet up to 65,536.
- Setting the maximum embryonic connections—The maximum number of embryonic connections per host up to 65,536. An embryonic connection is a connection request that has not finished the necessary handshake between source and destination. This limit enables the TCP intercept feature. (See the "Other Protection Features" section on page 1-6 for more information.)
- Disabling TCP sequence number randomization—Only use this option if another in-line firewall is also randomizing sequence numbers and the result is scrambling the data.

When you do not want to use NAT, such as for a transparent firewall or same security interfaces, you can set these options in an identity NAT statement or a NAT exemption statement.

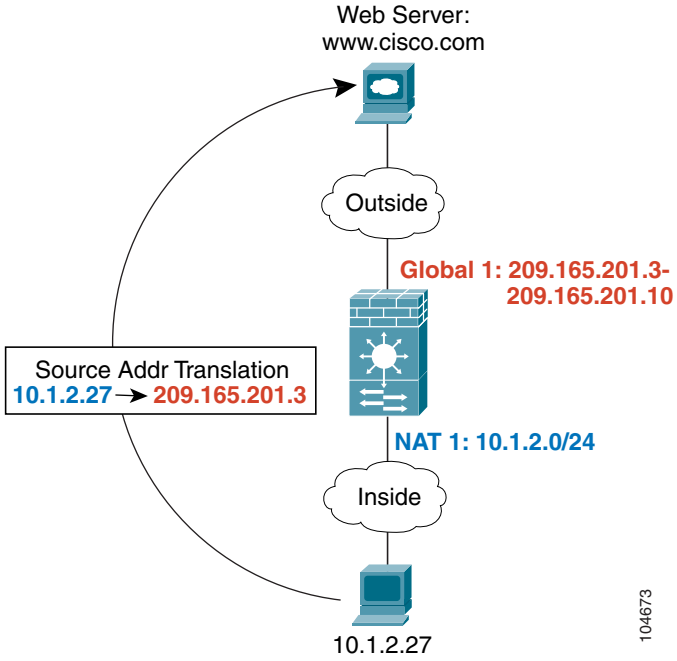
Using Dynamic NAT and PAT

This section includes the following topics:

Dynamic NAT and PAT Implementation

For dynamic NAT and PAT, you first configure a NAT statement (the **nat** command) identifying the addresses on a given interface that you want to translate. Then you configure a separate global statement (the **global** command) to specify the translated addresses when exiting another interface (in the case of PAT, this is one address). Each NAT statement matches a global statement by comparing the NAT ID, a number that you assign each statement (see Figure 9-8).

Figure 9-8 NAT and Global ID Matching

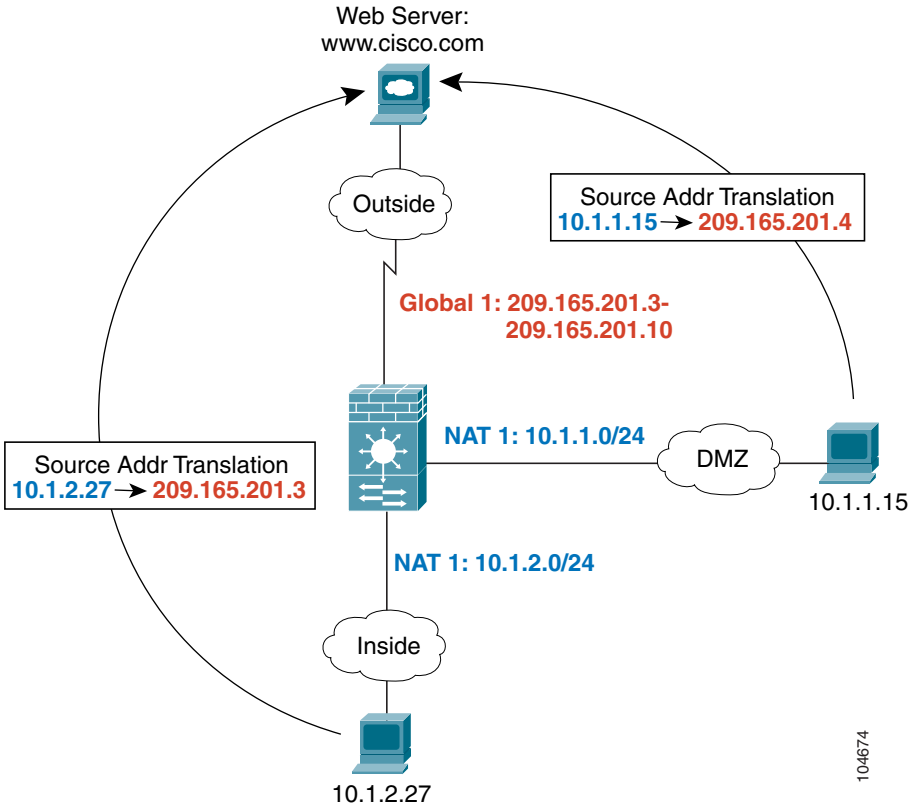


See the following commands for this example:

```
FWSM/contexta(config)# nat (inside) 1 10.1.2.0 255.255.255.0
FWSM/contexta(config)# global (outside) 1 209.165.201.3-209.165.201.10
```

You can enter a NAT statement for each interface using the same NAT ID; they all use the same global statement when traffic exits a given interface. For example, you can configure NAT statements for Inside and DMZ interfaces, both on NAT ID 1. Then you configure a global statement on the Outside interface that is also on ID 1. Traffic from the Inside interface and the DMZ interface share a NAT pool or a PAT address when exiting the Outside interface (see Figure 9-9).

Figure 9-9 NAT Statements on Multiple Interfaces

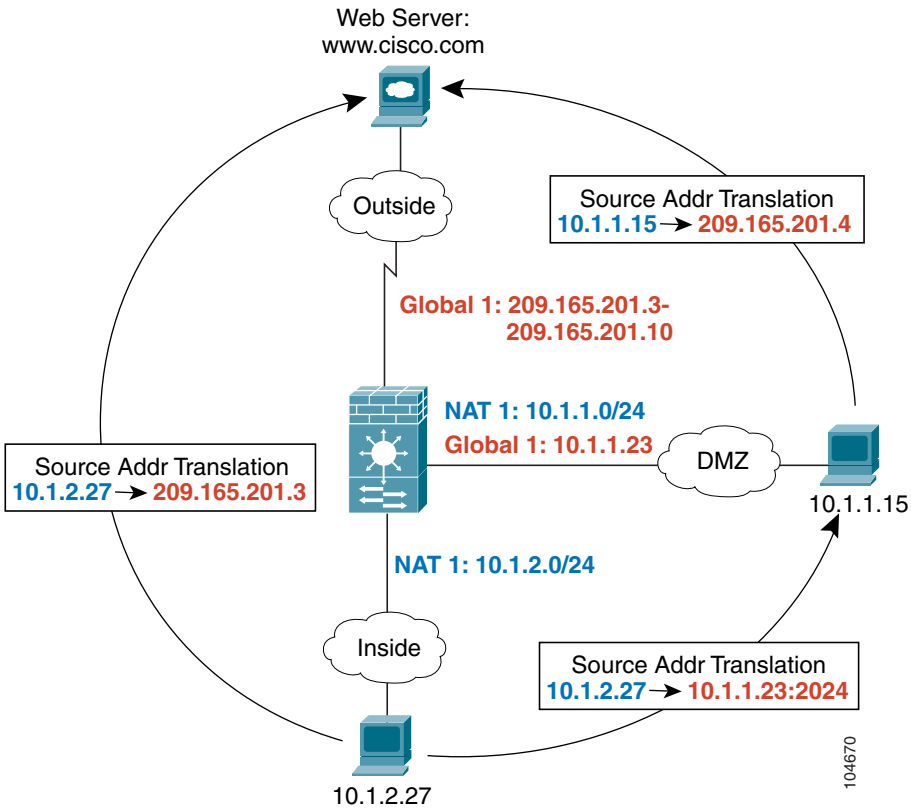


See the following commands for this example:

```
FWSM/contexta(config)# nat (inside) 1 10.1.2.0 255.255.255.0
FWSM/contexta(config)# nat (dmz) 1 10.1.1.0 255.255.255.0
FWSM/contexta(config)# global (outside) 1 209.165.201.3-209.165.201.10
```

You can also enter a global statement for each interface using the same NAT ID. If you enter a global statement for the Outside and DMZ interfaces on ID 1, then the Inside NAT statement identifies traffic to be translated when going to both the Outside and the DMZ interfaces. Similarly, if you also enter a NAT statement for the DMZ interface on ID 1, then the global statement on the Outside interface is also used for DMZ traffic. (See Figure 9-10).

Figure 9-10 Global and NAT Statements on Multiple Interfaces

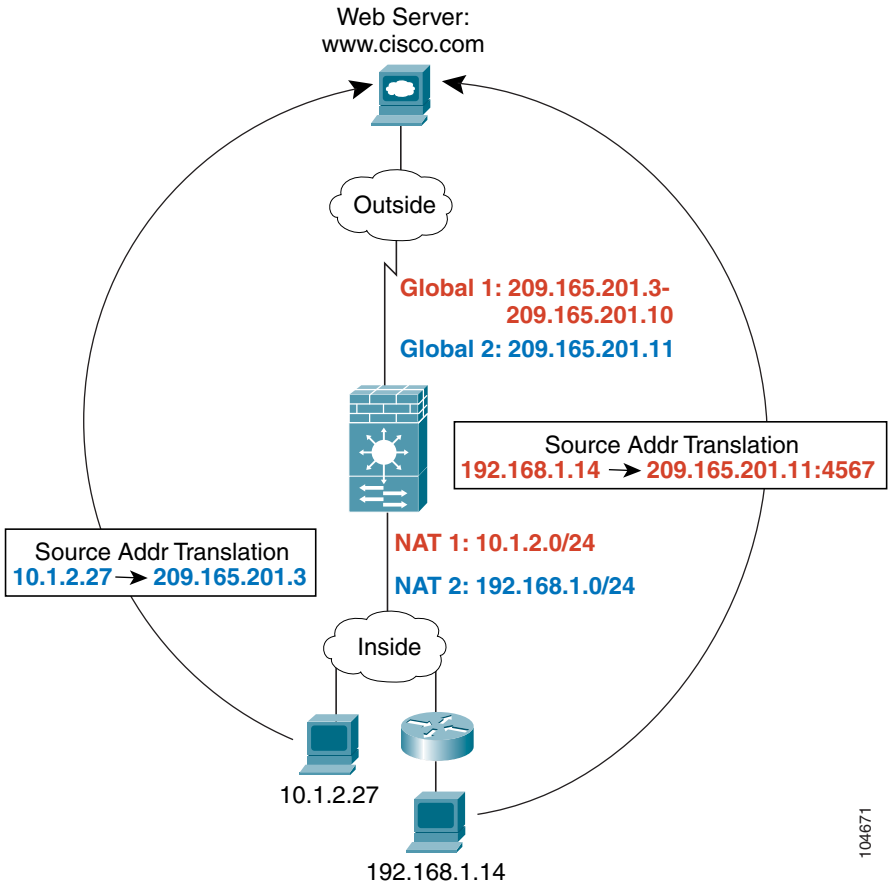


See the following commands for this example:

```
FWSM/contexta(config)# nat (inside) 1 10.1.2.0 255.255.255.0
FWSM/contexta(config)# nat (dmz) 1 10.1.1.0 255.255.255.0
FWSM/contexta(config)# global (outside) 1 209.165.201.3-209.165.201.10
FWSM/contexta(config)# global (dmz) 1 10.1.1.23
```

If you use different NAT IDs, you can identify different sets of host addresses to have different global addresses. For example, on the Inside interface, you can have two NAT statements on two different NAT IDs. On the Outside interface, you configure two global statements for these two IDs. Then, when traffic from Inside network A exits the Outside interface, the IP addresses are translated to pool A addresses; while traffic from Inside network B are translated to pool B addresses (see Figure 9-11). If you use policy NAT, you can specify the same local addresses for multiple NAT statements, as long as the source address/port and destination address/port is unique for each statement. For regular NAT, you must identify different local addresses for each statement.

Figure 9-11 Different NAT IDs

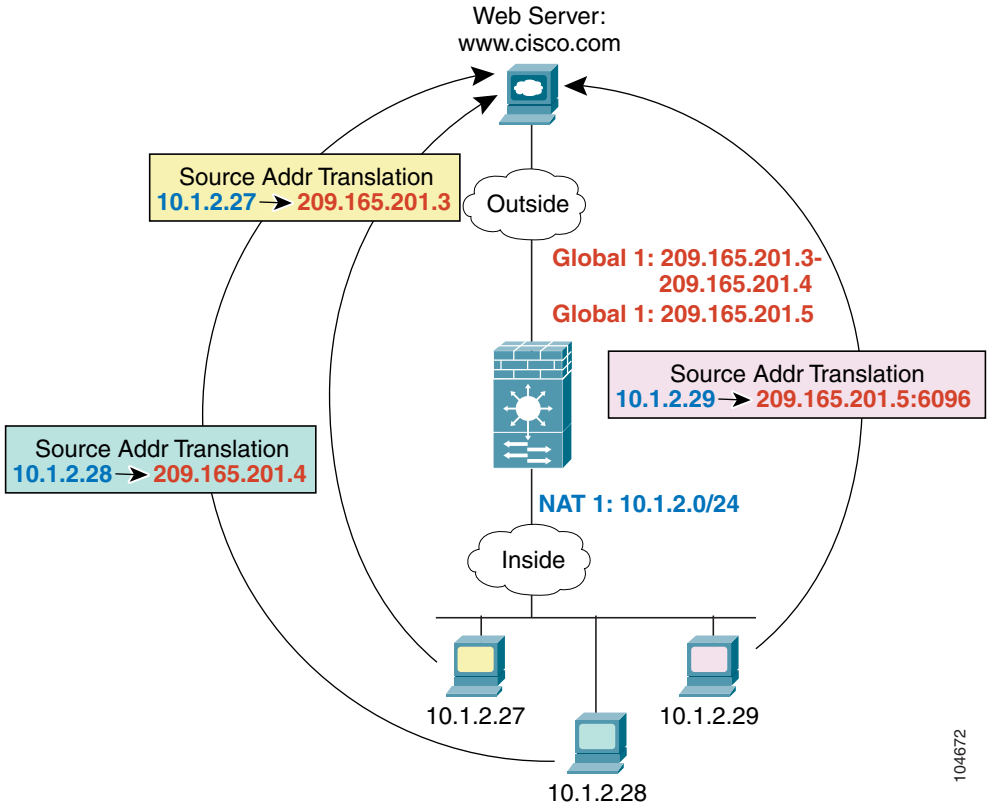


See the following commands for this example:

```
FWSM/contexta(config)# nat (inside) 1 10.1.2.0 255.255.255.0
FWSM/contexta(config)# nat (inside) 2 192.168.1.0 255.255.255.0
FWSM/contexta(config)# global (outside) 1 209.165.201.3-209.165.201.10
FWSM/contexta(config)# global (outside) 2 209.165.201.11
```

You can enter multiple global statements for one interface using the same NAT ID; the FWSM uses the dynamic NAT global statements first, in the order they are in the configuration, and then uses the PAT global statements in order. You might want to enter both a dynamic NAT global statement and a PAT global statement if you need to use dynamic NAT for a particular application, but want to have a backup PAT statement in case all the dynamic NAT addresses are used up. Similarly, you might enter two PAT statements if you need more than the approximately 64000 connections that a single PAT global statement supports (see Figure 9-12).

Figure 9-12 NAT and PAT Together



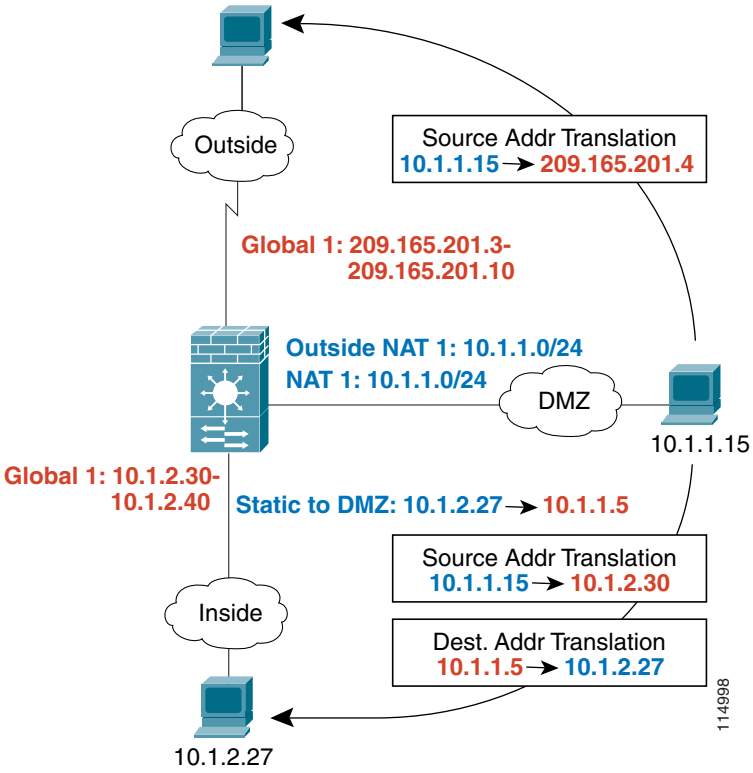
See the following commands for this example:

```
FWSM/contexta(config)# nat (inside) 1 10.1.2.0 255.255.255.0
FWSM/contexta(config)# global (outside) 1 209.165.201.3-209.165.201.4
FWSM/contexta(config)# global (outside) 1 209.165.201.5
```

For outside NAT (see the "Outside NAT" section on page 9-10 for more information), you need to identify the NAT statement for outside NAT (the **outside** keyword). If you also want to translate the same traffic when it accesses an inside interface (for example, traffic on a DMZ is translated when accessing the Inside and the Outside interfaces), then you must configure a separate NAT statement without the **outside** option. In this case, you can identify the same addresses in both statements and use the same

NAT ID (see Figure 9-13). Note that for outside NAT (DMZ interface to Inside interface), the inside host uses a static NAT statement to allow outside access, so both the source and destination addresses are translated.

Figure 9-13 Outside NAT and Inside NAT Combined



See the following commands for this example:

```
FWSM/contexta(config)# nat (dmz) 1 10.1.1.0 255.255.255.0 outside
FWSM/contexta(config)# nat (dmz) 1 10.1.1.0 255.255.255.0
FWSM/contexta(config)# static (inside,dmz) 10.1.2.27 10.1.1.5 netmask 255.255.255.255
FWSM/contexta(config)# global (outside) 1 209.165.201.3-209.165.201.4
FWSM/contexta(config)# global (inside) 1 10.1.2.30-1-10.1.2.40
```

Configuring NAT or PAT

This section tells how to configure dynamic NAT or dynamic PAT. The configuration for dynamic NAT and PAT are almost identical; for NAT you specify a range of global addresses, and for PAT you specify a single address.

Figure 9-14 shows a typical dynamic NAT scenario. Only local traffic can originate connections, and the global address is dynamically assigned from a pool.

Figure 9-14 Dynamic NAT

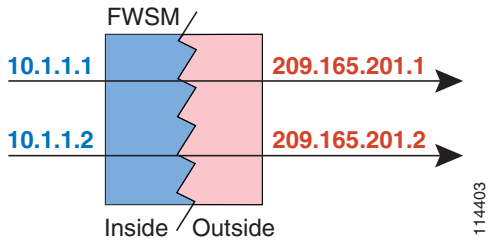
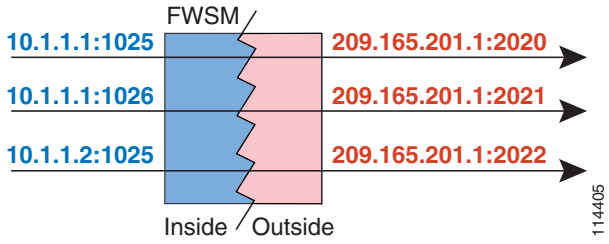


Figure 9-15 shows a typical dynamic PAT scenario. Only local traffic can originate connections, the global address is the same for each translation, but the port is dynamically assigned.

Figure 9-15 Dynamic PAT



For more information about dynamic NAT, see the "Dynamic NAT" section on page 9-3. For more information about PAT, see the "PAT" section on page 9-4.

Note If you change the NAT configuration, and you do not want to wait for existing translations to time out before the new NAT information is used, you can clear the translation table using the **clear xlate** command. However, clearing the translation table disconnects all current connections.

To configure dynamic NAT or PAT, follow these steps:

Step 1 To identify the local addresses that you want to translate, enter one of the following commands:

- Policy NAT:

```
FWSM/contexta(config)# nat (local_interface) nat_id access-list acl_name [dns]
[outside | [norandomseq] [[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns]]
```

You can identify overlapping addresses in other **nat** statements. For example, you can identify 10.1.1.0 in one statement, but 10.1.1.1 in another. The traffic is matched to a policy NAT statement in order, until the first match, or for regular NAT, using the best match.

See the following description about options for this command:

- **access-list** *acl_name*—Identify the local addresses and destination addresses using an extended ACL. Create the ACL using the **access-list** command (see the "Adding an Extended Access Control List" section on page 10-13). This ACL should include only **permit** access control entries (ACEs). You can optionally specify the local and destination ports in the ACL using the **eq** operator.
- *nat_id*—An integer between 1 and 65535. The NAT ID must match a **global** statement NAT ID. See the "Dynamic NAT and PAT Implementation" section on page 9-17 for more information about how NAT IDs are used. **0** is reserved for NAT exemption. (See the "Configuring NAT Exemption" section on page 9-31 for more information about NAT exemption.)
- **dns**—If your NAT statement includes the address of a host that has an entry in a DNS server, and the DNS server is on a different interface from a client, then the client and the DNS server need different addresses for the host; one needs the global address and one needs the local address. This option rewrites the address in the DNS reply to the client. The translated host needs to be on the same interface as either the client or the DNS server. Typically, hosts that need to allow access from other interfaces use a static translation, so this option is more likely to be used with the **static** command. (See the "DNS and NAT" section on page 9-13 for more information.)
- **outside**—If this interface is on a lower security level than the interface you identify by the matching **global** statement, then you must enter **outside** to identify the NAT instance as outside NAT. (See the "Outside NAT" section on page 9-10 for more information.)
- **norandomseq—**No TCP Initial Sequence Number (ISN) randomization. Only use this option if another in-line firewall is also randomizing sequence numbers and the result is scrambling the data. See the "Security Level Overview" section on page 6-6 for information about TCP sequence numbers.
- **tcp** *tcp_max_conns,* **udp** *udp_max_conns*—The maximum number of simultaneous TCP and/or UDP connections for the entire subnet up to 65,536. The default is 0 for both protocols, which means the maximum connections.
- *emb_limit*—The maximum number of embryonic connections per host up to 65,536. An embryonic connection is a connection request that has not finished the necessary handshake between source and destination. This limit enables the TCP Intercept feature. (See the "Other Protection Features" section on page 1-6 for more information.) The default is 0, which means the maximum embryonic connections. You must enter the **tcp** *tcp_max_conns* before you enter the *emb_limit*. If you want to use the default value for *tcp_max_conns*, but change the *emb_limit*, then enter **0** for *tcp_max_conns*. Not supported for outside NAT.

- Regular NAT:

```
FWSM/contexta(config)# nat (local_interface) nat_id local_ip [mask [dns] [outside |
[norandomseq] [[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns]]]
```

The *nat_id* is an integer between 1 and 2147483647. The NAT ID must match a **global** statement NAT ID. See the "Dynamic NAT and PAT Implementation" section on page 9-17 for more information about how NAT IDs are used. **0** is reserved for identity NAT. See the "Configuring Identity NAT" section on page 9-29 for more information about identity NAT.

See the policy NAT command above for information about other options.

Step 2 To identify the global address(es) to which you want to translate the local addresses when they exit a particular interface, enter the following command:

```
FWSM/contexta(config)# global (global_interface) nat_id {global_ip[-global_ip] |
interface}
```

This NAT ID must match a **nat** statement NAT ID. The matching **nat** statement identifies the addresses that you want to translate when they exit this interface.

You can specify a single address (for PAT) or a range of addresses (for NAT). The range can go across subnet boundaries if desired. For example, you can specify the following "supernet":

```
192.168.1.1-192.168.2.254
```

For example, to translate the 10.1.1.0/24 network on the inside interface, and to change the embryonic limit, enter the following command. You must specify the **tcp** *tcp_max_conns* before specifying *emb_limit*, so the command enters the default setting of **0** for *tcp_max_conns*.

```
FWSM/contexta(config)# nat (inside) 1 10.1.1.0 255.255.255.0 tcp 0 200
FWSM/contexta(config)# global (outside) 1 209.165.201.1-209.165.201.30
```

To identify a pool of addresses for dynamic NAT as well as a PAT address for when the NAT pool is exhausted, enter the following commands:

```
FWSM/contexta(config)# nat (inside) 1 10.1.1.0 255.255.255.0 tcp 5000 1000 udp 5000
FWSM/contexta(config)# global (outside) 1 209.165.201.5
FWSM/contexta(config)# global (outside) 1 209.165.201.10-209.165.201.20
```

To translate the lower security dmz network addresses so they appear to be on the same network as the inside network (10.1.1.0), for example, to simplify routing, enter the following commands:

```
FWSM/contexta(config)# nat (dmz) 1 10.1.2.0 255.255.255.0 outside dns
FWSM/contexta(config)# global (inside) 1 10.1.1.45
```

To identify a single local address with two different destination addresses using policy NAT, enter the following commands (see Figure 9-3 on page 9-8 for a related graphic):

```
FWSM/contexta(config)# access-list NET1 permit ip 10.1.2.0 255.255.255.0 209.165.201.0
255.255.255.224
FWSM/contexta(config)# access-list NET2 permit ip 10.1.2.0 255.255.255.0 209.165.200.224
255.255.255.224
FWSM/contexta(config)# nat (inside) 1 access-list NET1 tcp 0 2000 udp 10000
FWSM/contexta(config)# global (outside) 1 209.165.202.129
FWSM/contexta(config)# nat (inside) 2 access-list NET2 tcp 1000 500 udp 2000
FWSM/contexta(config)# global (outside) 2 209.165.202.130
```

To identify a single local address/destination address pair that use different ports using policy NAT, enter the following commands (see Figure 9-4 on page 9-9 for a related graphic):

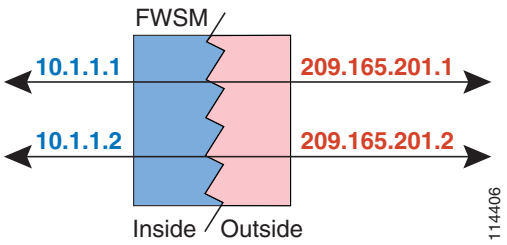
```
FWSM/contexta(config)# access-list WEB permit tcp 10.1.2.0 255.255.255.0 209.165.201.11
255.255.255.255 eq 80
FWSM/contexta(config)# access-list TELNET permit tcp 10.1.2.0 255.255.255.0 209.165.201.11
255.255.255.255 eq 23
FWSM/contexta(config)# nat (inside) 1 access-list WEB
FWSM/contexta(config)# global (outside) 1 209.165.202.129
FWSM/contexta(config)# nat (inside) 2 access-list TELNET
FWSM/contexta(config)# global (outside) 2 209.165.202.130
```

Using Static NAT

This section tells how to configure a static translation.

Figure 9-16 shows a typical static NAT scenario. Both local and global traffic can originate connections, and the global address is statically assigned.

Figure 9-16 Static NAT



You cannot use the same local or global address in multiple **static** statements between the same two interfaces. Do not use an address that is also defined as a dynamic PAT address in a **global** statement.

For more information about static NAT, see the "Static NAT" section on page 9-5.

Note If you change the NAT configuration, and you do not want to wait for existing translations to time out before the new NAT information is used, you can clear the translation table using the **clear xlate** command. However, clearing the translation table disconnects all current connections.

To configure static NAT, enter one of the following commands.

- For policy static NAT, enter the following command:

```
FWSM/contexta(config)# static (local_interface,global_interface)
{global_ip | interface} access-list acl_name [dns] [norandomseq] [[tcp] tcp_max_conns
[emb_limit]] [udp udp_max_conns]
```

  Create the ACL using the **access-list** command (see the "Adding an Extended Access Control List" section on page 10-13). This ACL should include only **permit** access control entries (ACEs). The source subnet mask used in the ACL is also used for the global addresses. You can also specify the local and destination ports in the ACL using the **eq** operator. See the "Policy NAT" section on page 9-8 for more information.

  See the "Configuring NAT or PAT" section on page 9-23 for information about the other options.

- To configure regular static NAT, enter the following command:

```
FWSM/contexta(config)# static (local_interface,global_interface)
{global_ip | interface} local_ip [netmask mask] [dns] [norandomseq]
[[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns]
```

  See the "Configuring NAT or PAT" section on page 9-23 for information about the options.

For example, the following policy static NAT example shows a single local address that is translated to two global addresses depending on the destination address (see Figure 9-3 on page 9-8 for a related graphic):

```
FWSM/contexta(config)# access-list NET1 permit ip host 10.1.2.27 209.165.201.0
255.255.255.224
FWSM/contexta(config)# access-list NET2 permit ip host 10.1.2.27 209.165.200.224
255.255.255.224
FWSM/contexta(config)# static (inside,outside) 209.165.202.129 access-list NET1
FWSM/contexta(config)# static (inside,outside) 209.165.202.130 access-list NET2
```

The following command maps an inside IP address (10.1.1.3) to an outside IP address (209.165.201.12):

```
FWSM/contexta(config)# static (inside,outside) 209.165.201.12 10.1.1.3 netmask
255.255.255.255
```

The following command maps the outside address (209.165.201.15) to an inside address (10.1.1.6):

```
FWSM/contexta(config)# static (outside,inside) 10.1.1.6 209.165.201.15 netmask
255.255.255.255
```

The following command statically maps an entire subnet:

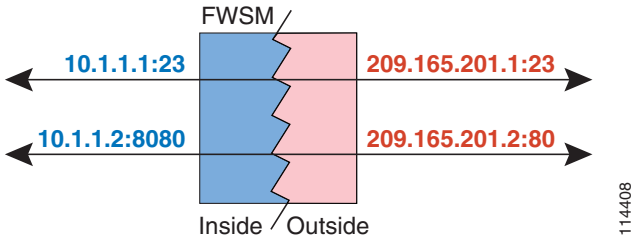
```
FWSM/contexta(config)# static (inside,dmz) 10.1.1.0 10.1.2.0 netmask 255.255.255.0
```

Using Static PAT

This section tells how to configure a static port translation. Static PAT lets you translate the local IP address to a global IP address, as well as the local port to a global port. You can choose to translate the same port, which lets you translate specific types of traffic, or you can take it further by translating to a different port.

Figure 9-17 shows a typical static PAT scenario. Both local and global traffic can originate connections, and the global address and port is statically assigned.

Figure 9-17 Static PAT



You cannot use the same local or global address in multiple **static** statements between the same two interfaces. Do not use an address that is also defined as a dynamic PAT address in a **global** statement.

For more information about static PAT, see the "Static PAT" section on page 9-5.

Note If you change the NAT configuration, and you do not want to wait for existing translations to time out before the new NAT information is used, you can clear the translation table using the **clear xlate** command. However, clearing the translation table disconnects all current connections.

To configure static PAT, enter one of the following commands.

- For policy static PAT, enter the following command:

```
FWSM/contexta(config)# static (local_interface,global_interface) {tcp | udp}
{global_ip | interface} global_port access-list acl_name [dns] [norandomseq]
[[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns]
```

  Create the ACL using the **access-list** command (see the "Adding an Extended Access Control List" section on page 10-13). The protocol in the ACL must match the protocol you set in this command. For example, if you specify **tcp** in the static command, then you must specify **tcp** in the ACL. Specify the port using the **eq** operator. This ACL should include only **permit** access control entries (ACEs). The source subnet mask used in the ACL is also used for the global addresses.

  See the "Configuring NAT or PAT" section on page 9-23 for information about the other options.

- To configure regular static PAT, enter the following command:

```
FWSM/contexta(config)# static (local_interface,global_interface) {tcp | udp}
{global_ip | interface} global_port local_ip local_port [netmask mask]
[dns] [norandomseq] [[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns]
```

  See the "Configuring NAT or PAT" section on page 9-23 for information about the options.

For example, for Telnet traffic initiated from hosts on the 10.1.3.0 network to the FWSM outside interface (10.1.2.14), you can redirect the traffic to the inside host at 10.1.1.15 by entering the following commands:

```
FWSM/contexta(config)# access-list TELNET permit tcp host 10.1.1.15 eq telnet 10.1.3.0
255.255.255.0 eq telnet
FWSM/contexta(config)# static (inside,outside) tcp 10.1.2.14 telnet access-list TELNET
```

For HTTP traffic initiated from hosts on the 10.1.3.0 network to the FWSM outside interface (10.1.2.14), you can redirect the traffic to the inside host at 10.1.1.15 by entering:

```
FWSM/contexta(config)# access-list HTTP permit tcp host 10.1.1.15 eq http 10.1.3.0
255.255.255.0 eq http
FWSM/contexta(config)# static (inside,outside) tcp 10.1.2.14 http access-list HTTP
```

To redirect Telnet traffic from the FWSM outside interface (10.1.2.14) to the inside host at 10.1.1.15, enter the following command:

```
FWSM/contexta(config)# static (inside,outside) tcp 10.1.2.14 telnet 10.1.1.15 telnet
netmask 255.255.255.255
```

If you want to allow the local Telnet server above to initiate connections, though, then you need to provide additional translation. For example, to translate all other types of traffic, enter the following commands. The original **static** command provides translation for Telnet to the server, while the **nat** and **global** commands provide PAT for outbound connections from the server.

```
FWSM/contexta(config)# static (inside,outside) tcp 10.1.2.14 telnet 10.1.1.15 telnet
netmask 255.255.255.255
FWSM/contexta(config)# nat (inside) 1 10.1.1.15 255.255.255.255
FWSM/contexta(config)# global (outside) 1 10.1.2.14
```

If you also have a separate translation for all inside traffic, and the inside hosts use a different global address from the Telnet server, you can still configure traffic initiated from the Telnet server to use the same global address as the **static** statement that allows Telnet traffic to the server. You need to create a more exclusive **nat** statement just for the Telnet server. Because **nat** statements are read for the best

match, more exclusive **nat** statements are matched before general statements. The following example shows the Telnet **static** statement, the more exclusive **nat** statement for initiated traffic from the Telnet server, and the statement for other inside hosts, which uses a different global address.

```
FWSM/contexta(config)# static (inside,outside) tcp 10.1.2.14 telnet 10.1.1.15 telnet
netmask 255.255.255.255
FWSM/contexta(config)# nat (inside) 1 10.1.1.15 255.255.255.255
FWSM/contexta(config)# global (outside) 1 10.1.2.14
FWSM/contexta(config)# nat (inside) 2 10.1.1.0 255.255.255.0
FWSM/contexta(config)# global (outside) 2 10.1.2.78
```

To translate a well-known port (80) to another port (8080), enter the following command:

```
FWSM/contexta(config)# static (inside,outside) tcp 10.1.2.45 80 10.1.1.16 8080 netmask
255.255.255.255
```

Bypassing NAT

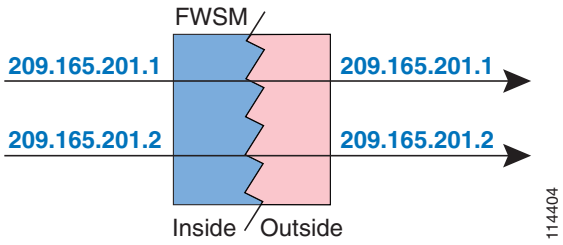
You can bypass NAT using identity NAT, static identity NAT, or NAT exemption. See the "Bypassing NAT" section on page 9-7 for more information about these methods. This section includes the following topics:

Configuring Identity NAT

Identity NAT translates the local IP address to the same IP address, and only local traffic can originate connections. (For same security level interfaces, hosts connected to any interface on the same security level can initiate traffic.)

Figure 9-18 shows a typical identity NAT scenario.

Figure 9-18 Identity NAT



Note If you change the NAT configuration, and you do not want to wait for existing translations to time out before the new NAT information is used, you can clear the translation table using the **clear xlate** command. However, clearing the translation table disconnects all current connections.

To configure identity NAT, enter the following command:

```
FWSM/contexta(config)# nat (local_interface) 0 local_ip [mask [dns] [outside |
[norandomseq] [[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns]]]
```

See the "Configuring NAT or PAT" section on page 9-23 for information about the options.

For example, to use identity NAT for the inside 10.1.1.0/24 network, enter the following command:

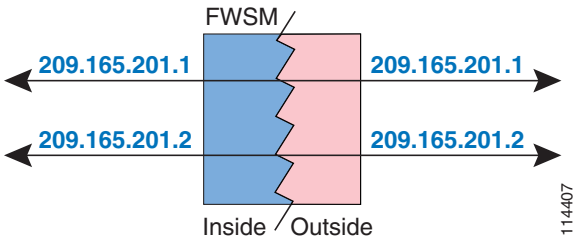
```
FWSM/contexta(config)# nat (inside) 0 10.1.1.0 255.255.255.0
```

Configuring Static Identity NAT

Static identity NAT translates the local IP address to the same IP address, and allows both local and global traffic to originate connections. Static identity NAT lets you use regular NAT or policy NAT. Policy NAT allow you to identify the local and destination addresses when determining the local traffic to translate (see the "Policy NAT" section on page 9-8 for more information about policy NAT). For example, you can use policy static identity NAT for an inside address when it accesses the outside interface and the destination is server A, but use a normal translation when accessing the outside server B.

Figure 9-19 shows a typical static identity NAT scenario.

Figure 9-19 Static Identity NAT



Note If you change the NAT configuration, and you do not want to wait for existing translations to time out before the new NAT information is used, you can clear the translation table using the **clear xlate** command. However, clearing the translation table disconnects all current connections.

To configure static identity NAT, enter one of the following commands:

- To configure policy static identity NAT, enter the following command:

  ```
  FWSM/contexta(config)# static (local_interface,global_interface) local_ip access-list
  acl_id [dns] [norandomseq] [[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns]
  ```

  Create the ACL using the **access-list** command (see the "Adding an Extended Access Control List" section on page 10-13). This ACL should include only **permit** access control entries (ACEs). Make sure the source address in the ACL matches the first *local_ip* in this command. See the "Policy NAT" section on page 9-8 for more information.

  See the "Configuring NAT or PAT" section on page 9-23 for information about the other options.

- To configure regular static identity NAT, enter the following command:

  ```
  FWSM/contexta(config)# static (local_interface,global_interface) local_ip local_ip
  [netmask mask] [dns] [norandomseq] [[tcp] tcp_max_conns [emb_limit]] [udp
  udp_max_conns]
  ```

Specify the same IP address for both *local_ip* variables.

See the "Configuring NAT or PAT" section on page 9-23 for information about the other options.

For example, the following command uses static identity NAT for an inside IP address (10.1.1.3) when accessed by the outside:

```
FWSM/contexta(config)# static (inside,outside) 10.1.1.3 10.1.1.3 netmask 255.255.255.255
```

The following command uses static identity NAT for an outside address (209.165.201.15) when accessed by the inside:

```
FWSM/contexta(config)# static (outside,inside) 209.165.201.15 209.165.201.15 netmask
255.255.255.255
```

The following command statically maps an entire subnet:

```
FWSM/contexta(config)# static (inside,dmz) 10.1.2.0 10.1.2.0 netmask 255.255.255.0
```

The following static identity policy NAT example shows a single local address that uses identity NAT when accessing one destination address, and a translation when accessing another:

```
FWSM/contexta(config)# access-list NET1 permit ip host 10.1.2.27 209.165.201.0
255.255.255.224
FWSM/contexta(config)# access-list NET2 permit ip host 10.1.2.27 209.165.200.224
255.255.255.224
FWSM/contexta(config)# static (inside,outside) 10.1.2.27 access-list NET1
FWSM/contexta(config)# static (inside,outside) 209.165.202.130 access-list NET2
```

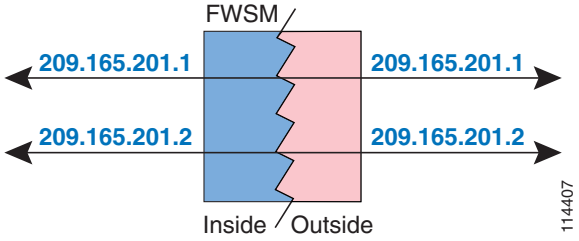
Configuring NAT Exemption

NAT exemption translates the local IP address to the same IP address, and allows both local and global traffic to originate connections. NAT exemption lets you specify the local and destination addresses when determining the local traffic to translate (similar to policy NAT), so you have greater control using NAT exemption than identity NAT. However unlike policy NAT, NAT exemption does not consider the ports in the ACL.

Note In multiple context mode, you cannot initiate connections from an interface shared between contexts when you use NAT exemption for the destination address. The classifier can only assign packets from a shared interface to a context when you configure a static statement for the destination address. For example, if you share the outside interface, you cannot use NAT exemption on an inside interface if you want outside traffic to reach the inside addresses. The classifier only looks at static statements where the global interface matches the source interface of the packet. Because NAT exemption does not identify a global interface, the classifier does not consider those NAT statements for classification purposes.

Figure 9-19 shows a typical NAT exemption scenario.

Figure 9-20 NAT Exemption



Note If you change the NAT configuration, and you do not want to wait for existing translations to time out before the new NAT information is used, you can clear the translation table using the **clear xlate** command. However, clearing the translation table disconnects all current connections.

To configure NAT exemption, enter the following command:

```
FWSM/contexta(config)# FWSM/contexta(config)# nat (local_interface) 0 access-list acl_name
[outside] [norandomseq] [[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns]
```

Create the ACL using the **access-list** command (see the "Adding an Extended Access Control List" section on page 10-13). This ACL should include only **permit** access control entries (ACEs). Do not specify the local and destination ports in the ACL; NAT exemption does not consider the ports.

See the "Configuring NAT or PAT" section on page 9-23 for information about the other options.

For example, to exempt an inside network when accessing any destination address, enter the following command:

```
FWSM/contexta(config)# access-list EXEMPT permit ip 10.1.2.0 255.255.255.0 any
FWSM/contexta(config)# nat (inside) 0 access-list EXEMPT
```

To exempt an inside address when accessing two different destination addresses, enter the following commands:

```
FWSM/contexta(config)# access-list NET1 permit ip 10.1.2.0 255.255.255.0 209.165.201.0
255.255.255.224
FWSM/contexta(config)# access-list NET1 permit ip 10.1.2.0 255.255.255.0 209.165.200.224
255.255.255.224
FWSM/contexta(config)# nat (inside) 0 access-list NET1
```

NAT Examples

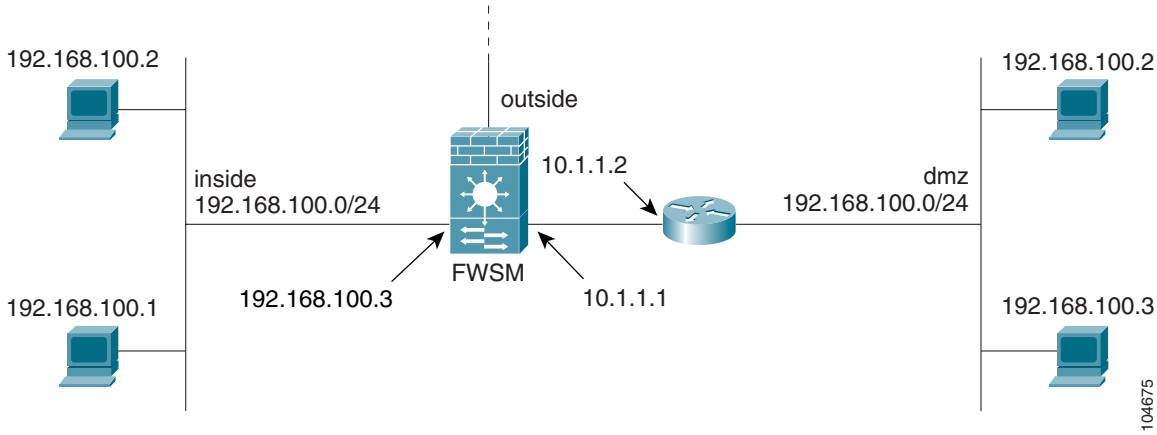
The following sections show typical scenarios that use NAT solutions:

- Overlapping Networks, page 9-33
- Redirecting Ports, page 9-34

Overlapping Networks

In Figure 9-21, the FWSM connects two private networks with overlapping address ranges.

Figure 9-21 Using Outside NAT with Overlapping Networks



Two networks use an overlapping address space (192.168.100.0/24), but hosts on each network must communicate (as allowed by ACLs). Without NAT, when a host on the inside network tries to access a host on the overlapping dmz network, the packet never makes it past the FWSM, which sees the packet as having a destination address on the inside network. Moreover, if the destination address is being used by another host on the inside network, that host receives the packet.

To solve this problem, use NAT to provide non-overlapping addresses. If you want to allow access in both directions, use static NAT for both networks. If you only want to allow the inside interface to access hosts on the dmz, then you can use dynamic NAT for the inside addresses, and static NAT for the dmz addresses you want to access. This example shows static NAT.

To configure static NAT for these two interfaces, follow these steps. The 10.1.1.0/24 network on the dmz is not translated.

Step 1 Translate 192.168.100.0/24 on the inside to 10.1.2.0 /24 when it accesses the dmz by entering the following command:

```
FWSM/contexta(config)# static (inside,dmz) 10.1.2.0 192.168.100.0 netmask 255.255.255.0
```

Step 2 Translate the 192.168.100.0/24 network on the dmz to 10.1.3.0/24 when it accesses the inside by entering the following command:

```
FWSM/contexta(config)# static (dmz,inside) 10.1.3.0 192.168.100.0 netmask 255.255.255.0
```

Step 3 Configure the following static routes so that traffic to the dmz network can be routed correctly by the FWSM:

```
FWSM/contexta(config)# route dmz 192.168.100.128 255.255.255.128 10.1.1.2 1
FWSM/contexta(config)# route dmz 192.168.100.0 255.255.255.128 10.1.1.2 1
```

The FWSM already has a connected route for the inside network. These static routes allow the FWSM to send traffic for the 192.168.100.0/24 network out the dmz interface to the gateway router at 10.1.1.2. (You need to split the network into two because you cannot create a static route with the exact same network as a connected route.) Alternatively, you could use a more broad route for the dmz traffic, such as a default route.

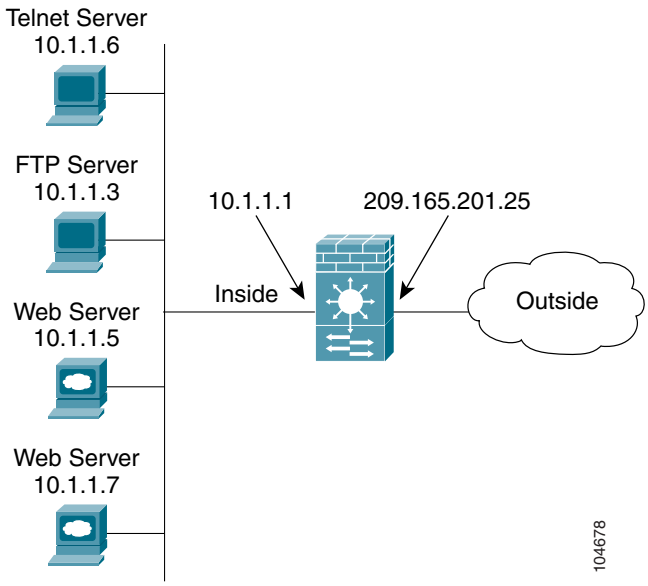
If host 192.168.100.2 on the dmz network wants to initiate a connection to host 192.168.100.2 on the inside network, the following events occur:

1. The dmz host 192.168.100.2 sends the packet to IP address 10.1.2.2.
2. When the FWSM receives this packet, the FWSM translates the source address from 192.168.100.2 to 10.1.3.2.
3. Then the FWSM translates the destination address from 10.1.2.2 to 192.168.100.2, and the packet is forwarded.

Redirecting Ports

Figure 9-22 illustrates a typical network scenario in which the port redirection feature might be useful.

Figure 9-22 Port Redirection Using Static PAT



In the configuration described in this section, port redirection occurs for hosts on external networks as follows:

- Telnet requests to IP address 209.165.201.5 are redirected to 10.1.1.6
- FTP requests to IP address 209.165.201.5 are redirected to 10.1.1.3
- HTTP request to FWSM outside IP address 209.165.201.25 are redirected to 10.1.1.5
- HTTP port 8080 requests to PAT address 209.165.201.15 are redirected to 10.1.1.7 port 80

To implement this scenario, complete the following steps:

Step 1 Configure PAT for the inside network by entering the following commands:

```
FWSM/contexta(config)# nat (inside) 1 0.0.0.0 0.0.0.0 0 0
FWSM/contexta(config)# global (outside) 1 209.165.201.15
```

Step 2 Redirect Telnet requests for 209.165.201.5 to 10.1.1.6 by entering the following command:

```
FWSM/contexta(config)# static (inside,outside) tcp 209.165.201.5 telnet 10.1.1.6 telnet
netmask 255.255.255.255
```

Step 3 Redirect FTP requests for IP address 209.165.201.5 to 10.1.1.3 by entering the following command:

```
FWSM/contexta(config)# static (inside,outside) tcp 209.165.201.5 ftp 10.1.1.3 ftp netmask
255.255.255.255
```

Step 4 Redirect HTTP requests for the FWSM outside interface address to 10.1.1.5 by entering the following command:

```
FWSM/contexta(config)# static (inside,outside) tcp interface www 10.1.1.5 www netmask
255.255.255.255
```

Step 5 Redirect HTTP requests on port 8080 for PAT address 209.165.201.15 to 10.1.1.7 port 80 by entering the following command:

```
FWSM/contexta(config)# static (inside,outside) tcp 209.165.201.15 8080 10.1.1.7 www
netmask 255.255.255.255
```

CHAPTER 10

Controlling Network Access with Access Control Lists

This chapter tells how to control network access through the Firewall Services Module (FWSM) using access control lists (ACLs). You can also use ACLs for other purposes, for example, to identify addresses for NAT, AAA, or OSPF route redistribution. This chapter describes how to create ACLs for these purposes as well as for network access, but this chapter only describes how to *apply* the ACLs for network access. Refer to the NAT, AAA, or IP chapters for information about applying ACLs for these other purposes.

Note You use ACLs to control network access in both routed and transparent firewall modes. In transparent mode, you can use both extended ACLs (for Layer 3 traffic) and EtherType ACLs (for Layer 2 traffic).

This chapter contains the following sections:

Access Control List Overview

ACLs are made up of one or more Access Control Entries (ACEs). An ACE is a single entry in an ACL that specifies a permit or deny rule, and is applied to a protocol, a source and destination IP address or network, and optionally the source and destination ports.

This section includes the following topics:

Access Control List Types and Uses

This section includes the following topics:

Access Control List Type Overview

Table 10-1 lists the types of ACLs you can create and how you can use them.

Table 10-1 Access Control List Types and Uses

ACL Use	ACL Type	For more information...
Control network access for IP traffic	Extended	See the "Controlling Network Access for IP Traffic (Extended)" section on page 10-2.
Identify traffic for AAA rules	Extended	See the "Identifying Traffic for AAA rules (Extended)" section on page 10-3.
Control network access for IP traffic for a given user	Extended, downloaded from a AAA server per user	See the "Controlling Network Access for IP Traffic for a Given User (Extended)" section on page 10-4.
Identify addresses for NAT (policy NAT and NAT exemption)	Extended	See the "Identifying Addresses for Policy NAT and NAT Exemption (Extended)" section on page 10-4.
Establish VPN management access	Extended	See the "VPN Management Access (Extended)" section on page 10-5.
For transparent firewall mode, control network access for non-IP traffic	EtherType	See the "Controlling Network Access for Non-IP Traffic (EtherType)" section on page 10-5.
Identify OSPF route redistribution	Standard	See the "Redistributing OSPF Routes (Standard)" section on page 10-6.

Controlling Network Access for IP Traffic (Extended)

Extended ACLs control connections based on source address, destination address, protocol, or port. The FWSM does not allow any traffic through unless it is explicitly permitted by an extended ACL. This rule is true for both routed firewall mode and transparent firewall mode.

For TCP and UDP connections, you do not need an ACL to allow returning traffic, because the FWSM allows all returning traffic for established connections. See the "Stateful Inspection Feature" section on page 1-5 for more information. For connectionless protocols such as ICMP, however, you either need

ACLs to allow ICMP in both directions (by applying ACLs to the source and destination interfaces), or you need to enable the ICMP inspection engine (see the "ICMP Inspection Engine" section on page 13-10). The ICMP inspection engine treats ICMP sessions as stateful connections.

You can apply one ACL of each type to each direction of an interface. You can also apply the same ACLs on multiple interfaces.

To control network access for IP traffic, perform the following task:

- Create and apply the ACL according to the "Adding an Extended Access Control List" section on page 10-13.

Allowing Special Traffic through the Transparent Firewall

In routed firewall mode, some types of traffic are blocked even if you allow them in an ACL, including unsupported dynamic routing protocols, DHCP (unless you configure DHCP relay), and multicast traffic. Transparent firewall mode can allow any IP traffic through. Because these special types of traffic are connectionless, you need to apply an ACL to both interfaces, so returning traffic is allowed through.

Table 10-2 lists common traffic types that you can allow through the transparent firewall. See Appendix D, "Addresses, Protocols, and Ports Reference," for more protocols and ports.

Table 10-2 Transparent Firewall Special Traffic

Traffic Type	Protocol or Port	Notes
BGP[1]	TCP port 179	—
DHCP[2]	UDP ports 67 and 68	If you enable the DHCP server, then the FWSM does not pass DHCP packets.
EIGRP[3]	Protocol 88	—
Multicast streams	The UDP ports vary depending on the application.	Multicast streams are always destined to a Class D address (224.0.0.0 to 239.x.x.x).
OSPF	Protocol 89	—
RIP (v1 or v2)	UDP port 520	—

1. Border Gateway Protocol
2. Dynamic Host Configuration Protocol
3. Enhanced Interior Gateway Routing Protocol

Identifying Traffic for AAA rules (Extended)

ACLs can be used with AAA in several ways.

- To identify traffic for network access authorization using a TACACS+ server, perform the following tasks:
  a. Add the ACL using the "Adding an Extended Access Control List" section on page 10-13.

     Permit entries in the ACL mark matching traffic for authorization, while deny entries exclude matching traffic from authorization.
  b. Apply the ACL using the **aaa authorization match** command in the "Configuring TACACS+ Authorization" section on page 12-22.

- To identify traffic for network access authentication using a TACACS+ or RADIUS server, perform the following tasks:
  a. Add the ACL using the "Adding an Extended Access Control List" section on page 10-13.

     Permit entries in the ACL mark matching traffic for authentication, while deny entries exclude matching traffic from authentication.
  b. Apply the ACL using the **aaa authentication match** command in the "Configuring Authentication for Network Access" section on page 12-20.
- To identify traffic for network access accounting using a TACACS+ or RADIUS server, perform the following tasks:
  a. Add the ACL using the "Adding an Extended Access Control List" section on page 10-13.

     Permit entries in the ACL mark matching traffic for accounting, while deny entries exclude matching traffic from accounting.
  b. Apply the ACL using the **aaa accounting match** command in the "Configuring Accounting for Network Access" section on page 12-25.

Controlling Network Access for IP Traffic for a Given User (Extended)

When you configure user authentication for network access, you can also choose to configure user authorization that determines the specific access privileges for each user. If you use a RADIUS server, you can configure the RADIUS server to download a dynamic ACL to be applied to the user, or the server can send the name of an ACL that you already configured on the FWSM. See the following tasks for each method.

- For dynamic ACLs, all ACL configuration takes place on the RADIUS server. Perform the following tasks:
  a. Refer to the "Adding an Extended Access Control List" section on page 10-13 for ACL syntax and guidelines.
  b. To create the ACL on the RADIUS server, see the "Configuring the RADIUS Server to Download Per-User Access Control Lists" section on page 12-23.
- For a downloaded ACL name, perform the following tasks:
  a. Configure an extended ACL according to the "Adding an Extended Access Control List" section on page 10-13.

     This extended ACL is not assigned to an interface, but is designed to be applied to one or more users.
  b. Use the ACL name according to the "Configuring the RADIUS Server to Download Per-User Access Control List Names" section on page 12-25.

These per-user ACLs must be as restrictive or more restrictive than an extended ACL that is assigned to the interface. For example, if the ACL assigned to the inside interface allows all users to have only HTTP access to other networks, it would not make sense to configure an authorization ACL for that user to access FTP.

Identifying Addresses for Policy NAT and NAT Exemption (Extended)

Policy NAT lets you identify local traffic for address translation by specifying the source and destination addresses in an extended ACL. You can also optionally specify the source and destination ports. Regular NAT can only consider the local addresses.

NAT exemption statements also use ACLs, but you cannot specify the ports.

To use ACLs with NAT, perform the following tasks:

1. Add the ACL using the "Adding an Extended Access Control List" section on page 10-13. This ACL can contain only permit elements. Specify ports using the **eq** operator.
2. Use the ACL in the **nat** and **static** commands described in the following sections:
   - "Using Dynamic NAT and PAT" section on page 9-15
   - "Using Static NAT" section on page 9-25
   - "Using Static PAT" section on page 9-26
   - "Configuring Static Identity NAT" section on page 9-29
   - "Configuring NAT Exemption" section on page 9-30

VPN Management Access (Extended)

You can use an extended ACL in VPN commands. See the following tasks for each method.

- To identify hosts allowed to connect to the FWSM over an IPSec site-to-site tunnel, perform the following tasks:
  a. Add the ACL using the "Adding an Extended Access Control List" section on page 10-13.

     Specify the FWSM address as the source address. Specify the remote address(es) for the destination address.

  b. Use the ACL in the **crypto map match address** command according to the "Configuring a Site-to-Site Tunnel" section on page 11-8.
- To identify the traffic that should be tunneled from a VPN client, perform the following tasks:
  a. Add the ACL using the "Adding an Extended Access Control List" section on page 10-13.

     Specify the FWSM address as the source address, and the VPN pool addresses as the destination addresses.

  b. Then use the ACL in the **vpngroup split-tunnel** command according to the "Configuring VPN Client Access" section on page 11-7.

The FWSM only supports IPSec tunnels that terminate on the FWSM and that allow access to the FWSM for management purposes; you cannot terminate a tunnel on the FWSM for traffic that goes through the FWSM to another network.

Controlling Network Access for Non-IP Traffic (EtherType)

Transparent firewall mode only

You can configure an ACL that controls traffic based on its EtherType. The FWSM can control any EtherType identified by a 16-bit hexadecimal number. EtherType ACLs support Ethernet V2 frames. 802.3-formatted frames are not handled by the ACL because they use a length field as opposed to a type field. Bridge protocol data units (BPDUs), which are handled by the ACL, are the only exception: they are SNAP-encapsulated, and the FWSM is designed to specifically handle BPDUs.

To control non-IP traffic, perform the following task:

- Create and apply the ACL according to the "Adding an EtherType Access Control List" section on page 10-16.

Redistributing OSPF Routes (Standard)

Single context mode only

Standard ACLs include only the destination address. You can use a standard ACL with the **route-map** command to control the redistribution of OSPF routes, perform the following tasks:

1. Create the ACL according to the "Adding a Standard Access Control List" section on page 10-17.
2. Create a route map and apply it according to the "Redistributing Routes Between OSPF Processes" section on page 8-6.

Access Control List Guidelines

See the following guidelines for creating ACLs:

- Access Control Entry Order, page 10-6
- Access Control List Implicit Deny, page 10-6
- Access Control List Commit, page 10-6
- Maximum Number of ACEs, page 10-7
- IP Addresses Used for Access Control Lists When You Use NAT, page 10-7
- Inbound and Outbound Access Control Lists, page 10-10

Access Control Entry Order

An ACL is made up of one or more Access Control Entries (ACEs). Depending on the ACL type, you can specify the source and destination addresses, the protocol, the ports (for TCP or UDP), the ICMP type (for ICMP), or the EtherType.

Each ACE that you enter for a given ACL name is appended to the end of the ACL.

The order of ACEs is important. When the FWSM decides whether to forward or drop a packet, the FWSM tests the packet against each ACE in the order in which the entries are listed. After a match is found, no more ACEs are checked. For example, if you create an ACE at the beginning of an ACL that explicitly permits all traffic, no further statements are ever checked.

Access Control List Implicit Deny

ACLs have an implicit deny at the end of the list, so unless you explicitly permit it, traffic cannot pass. For example, if you want to allow all users to access a network through the FWSM except for particular addresses, then you need to deny the particular addresses and then permit all others.

Access Control List Commit

When you add an ACE to an ACL, the FWSM activates the ACL by committing it to the network processors. The FWSM waits a short period of time after you last entered an **access-list** command and then commits the ACL. This waiting period minimizes the number of times the FWSM commits the ACL. If you enter multiple ACEs within the short waiting period, or paste ACEs at the command prompt, then the FWSM does not commit the ACL until the waiting period has passed and you do not enter more entries. The FWSM displays a message similar to the following after it commits the ACL:

```
Access Rules Download Complete: Memory Utilization: < 1%
```

Large ACLs of approximately 60K ACEs can take 3 to 4 minutes to commit, depending on the size.

To manually commit ACLs, see the "Manually Committing Access Control Lists and Rules" section on page 10-24.

For information about exceeding memory limits, see the "Maximum Number of ACEs" section.

Maximum Number of ACEs

The FWSM supports a maximum of 80K rules for the entire system in single mode, and 142K rules for multiple mode. Rules include ACEs, ACEs used for policy NAT, filters, AAA, ICMP, Telnet, SSH, HTTP, and established rules. See the "Rule Limits" section on page A-5 for the limits for each rule type.

Some ACLs use more memory than others, and these include ACLs that use large port number ranges or overlapping networks (for example one ACE specifies 10.0.0.0/8 and another specifies 10.1.1.0/24). Depending on the type of ACL, the actual limit the system can support will be less than 80K (single mode) or 142K (multiple mode).

If you use object groups in ACEs, the number of actual ACEs that you enter is fewer, but the number of *expanded* ACEs is the same as without object groups, and expanded ACEs count towards the system limit. To view the number of expanded ACEs in an ACL, enter the **show access-list** *acl_name* command.

When you add an ACE, and the FWSM compiles the ACL, the console displays the memory used in a message similar to the following:

```
Access Rules Download Complete: Memory Utilization: < 1%
```

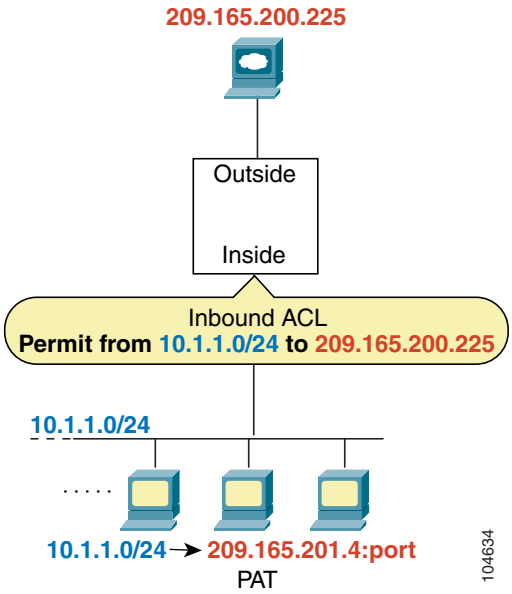
If you exceed the memory limitations, you receive an error message and a system message (106024), and all the ACLs that were added in this compilation are removed from the configuration. Only the set of ACLs that were successfully committed in the previous commitment are used. For example, if you paste 1,000 ACEs at the prompt, and the last ACE exceeds the memory limitations, all 1,000 ACEs are rejected.

IP Addresses Used for Access Control Lists When You Use NAT

When you use NAT, the IP addresses you specify for an ACL depend on the interface to which the ACL is attached; you need to use addresses that are valid on the network connected to the interface. This guideline applies for both inbound and outbound ACLs: the direction does not determine the address used, only the interface does.

For example, you want to apply an ACL to the inbound direction of the inside interface. You configure the FWSM to perform NAT on the inside source addresses when they access outside addresses. Because the ACL is applied to the inside interface, the source addresses are the original untranslated addresses. Because the outside addresses are not translated, the destination address used in the ACL is the real address (see Figure 10-1).

Figure 10-1 IP Addresses in ACLs: NAT Used for Source Addresses

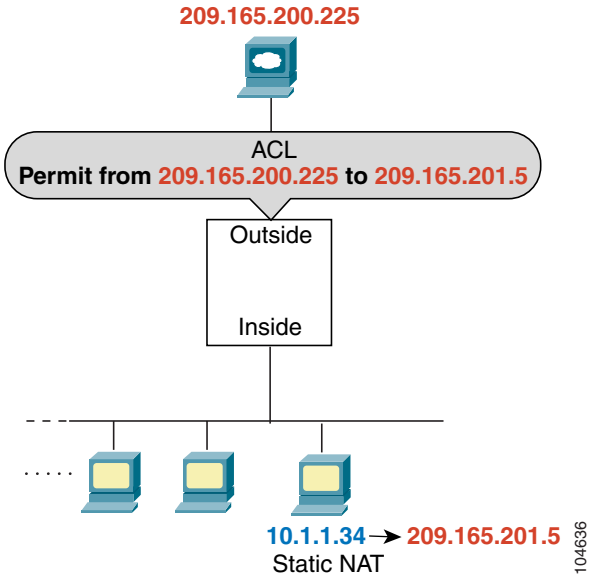


See the following commands for this example:

```
FWSM/contexta(config)# access-list INSIDE extended permit ip 10.1.1.0 255.255.255.0 host
209.165.200.225
FWSM/contexta(config)# access-group INSIDE in interface inside
```

If you want to allow an outside host to access an inside host, you can apply an inbound ACL on the outside interface. You need to specify the translated address of the inside host in the ACL because that address is the address that can be used on the outside network (see Figure 10-2).

Figure 10-2 IP Addresses in ACLs: NAT used for Destination Addresses

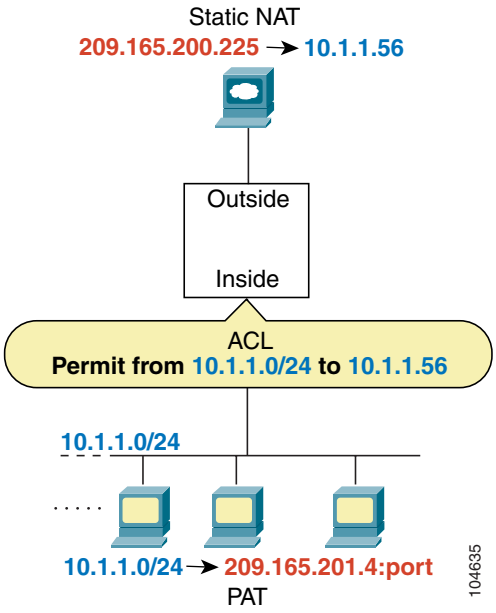


See the following commands for this example:

```
FWSM/contexta(config)# access-list OUTSIDE extended permit ip host 209.165.200.225 host
209.165.201.5
FWSM/contexta(config)# access-group OUTSIDE in interface outside
```

If you perform NAT on both interfaces, then keep in mind the addresses that are visible to a given interface. In Figure 10-3, an outside server uses static NAT so that a translated address appears on the inside network.

Figure 10-3 IP Addresses in ACLs: NAT used for Source and Destination Addresses



See the following commands for this example:

```
FWSM/contexta(config)# access-list INSIDE extended permit ip 10.1.1.0 255.255.255.0 host
10.1.1.56
FWSM/contexta(config)# access-group INSIDE in interface inside
```

For an example of IP addresses used in outbound ACLs, see Figure 10-5 on page 10-12.

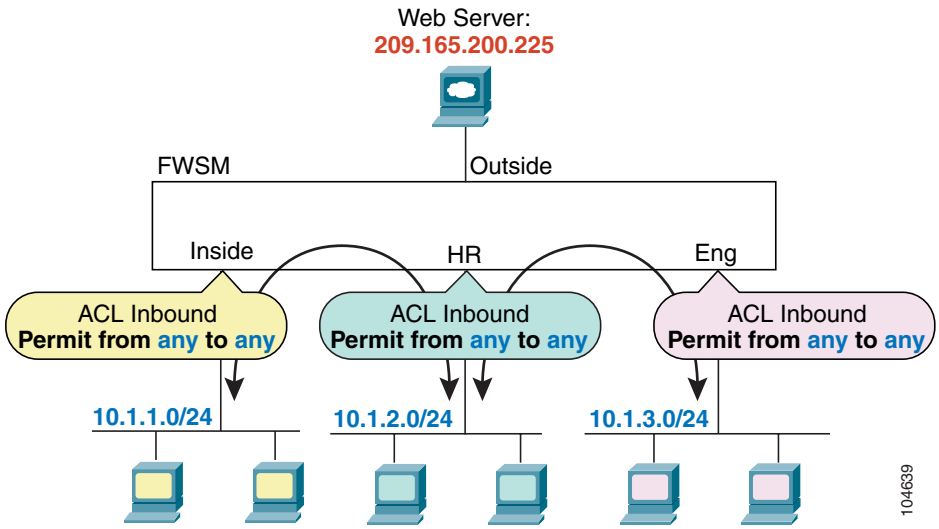
Inbound and Outbound Access Control Lists

Traffic flowing across an interface in the FWSM can be controlled in two ways. Traffic that enters the FWSM can be controlled by attaching an inbound ACL to the source interface. Traffic that exits the FWSM can be controlled by attaching an outbound ACL to the destination interface. To allow any traffic to enter the FWSM, you must attach an inbound ACL to an interface; otherwise, the FWSM automatically drops all traffic that enters that interface. By default, traffic can exit the FWSM on any interface unless you restrict it using an outbound ACL, which adds restrictions to those already configured in the inbound ACL.

Note "Inbound" and "outbound" refer to the application of an ACL on an interface, either to traffic entering the FWSM on an interface or traffic exiting the FWSM on an interface. These terms do not refer to the movement of traffic from a lower security interface to a higher security interface, commonly known as inbound, or from a higher to lower interface, commonly known as outbound.

You might want to use an outbound ACL to simplify your ACL configuration. For example, if you want to allow three inside networks on three different interfaces to access each other, you can create a simple inbound ACL that allows all traffic on each inside interface (see Figure 10-4).

Figure 10-4 Inbound ACLs



See the following commands for this example:

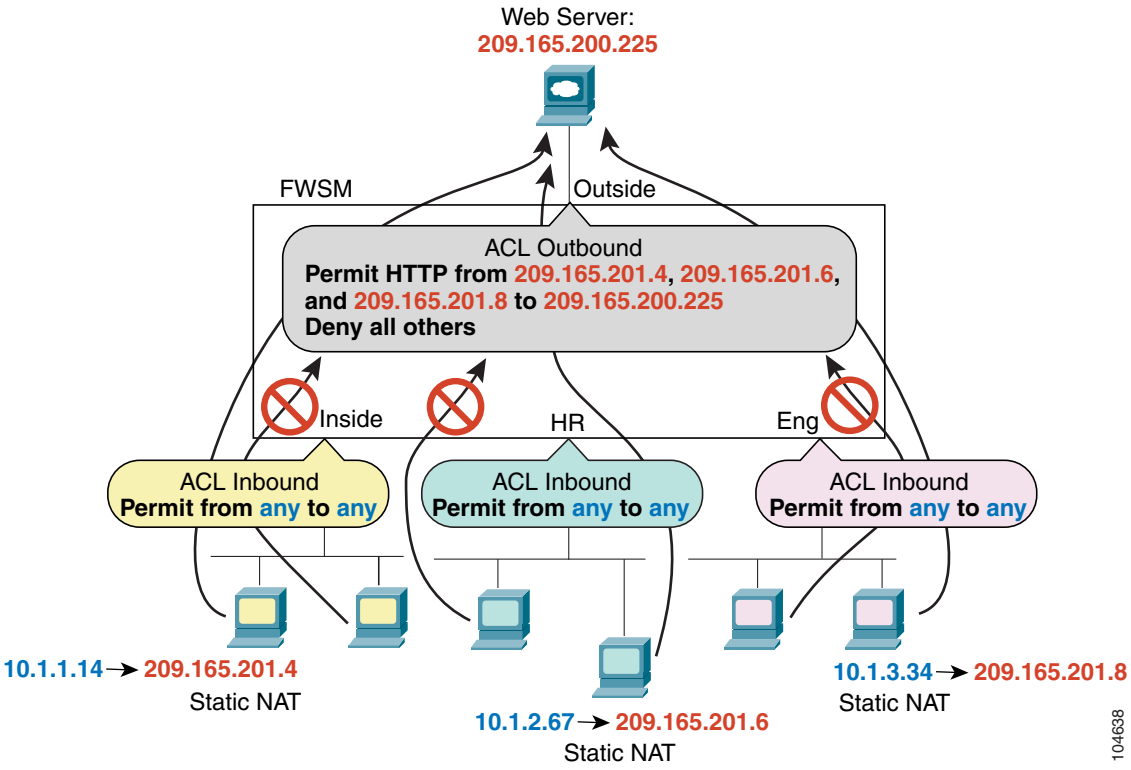
```
FWSM/contexta(config)# access-list INSIDE extended permit ip any any
FWSM/contexta(config)# access-group INSIDE in interface inside

FWSM/contexta(config)# access-list HR extended permit ip any any
FWSM/contexta(config)# access-group HR in interface hr

FWSM/contexta(config)# access-list ENG extended permit ip any any
FWSM/contexta(config)# access-group ENG in interface eng
```

Then, if you want to allow only certain hosts on the inside networks to access a web server on the outside network, you can create a more restrictive ACL that allows only the specified hosts and apply it to the outbound direction of the outside interface (see Figure 10-4). See the "IP Addresses Used for Access Control Lists When You Use NAT" section on page 10-7 for information about NAT and IP addresses. The outbound ACL prevents any other hosts from reaching the outside network.

Figure 10-5 Outbound ACL



See the following commands for this example:

```
FWSM/contexta(config)# access-list INSIDE extended permit ip any any
FWSM/contexta(config)# access-group INSIDE in interface inside

FWSM/contexta(config)# access-list HR extended permit ip any any
FWSM/contexta(config)# access-group HR in interface hr

FWSM/contexta(config)# access-list ENG extended permit ip any any
FWSM/contexta(config)# access-group ENG in interface eng

FWSM/contexta(config)# access-list OUTSIDE extended permit tcp host 209.165.201.4
host 209.165.200.225 eq www
FWSM/contexta(config)# access-list OUTSIDE extended permit tcp host 209.165.201.6
host 209.165.200.225 eq www
FWSM/contexta(config)# access-list OUTSIDE extended permit tcp host 209.165.201.8
host 209.165.200.225 eq www
FWSM/contexta(config)# access-group OUTSIDE out interface outside
```

Adding an Extended Access Control List

An extended ACL is made up of one or more ACEs, in which you can specify the source and destination addresses, and, depending on the ACE type, the protocol, the ports (for TCP or UDP), or the ICMP type (for ICMP). You can identify all of these parameters within the **access-list** command, or you can use object groups for each parameter. This section describes how to identify the parameters within the command. To use object groups, see the "Simplifying Access Control Lists with Object Grouping" section on page 10-17.

For TCP and UDP connections, you do not need to also apply an ACL on the destination interface to allow returning traffic, because the FWSM allows all returning traffic for established connections. See the "Stateful Inspection Feature" section on page 1-5 for more information. For connectionless protocols such as ICMP, however, you either need ACLs to allow ICMP in both directions (by applying ACLs to the source and destination interfaces), or you need to enable the ICMP inspection engine. (See the "ICMP Inspection Engine" section on page 13-10.) The ICMP inspection engine treats ICMP sessions as stateful connections. For transparent mode, you can allow protocols with an extended ACL that are otherwise blocked by a routed mode FWSM, including BGP, DHCP, and multicast streams. Because these protocols do not have sessions on the FWSM to allow returning traffic, these protocols also require ACLs on both interfaces.

You can apply only one ACL of each type (extended and EtherType) to each direction of an interface. You can apply the same ACLs on multiple interfaces.

Note If you change the ACL configuration, and you do not want to wait for existing connections to time out before the new ACL information is used, you can clear the translation table using the **clear xlate** command. However, clearing the translation table disconnects all current connections.

To add an extended ACL and apply it to an interface, follow these steps:

Step 1 Add one or more ACEs of the following types using the same ACL name.

When you enter the **access-list** command for a given ACL name, the ACE is added to the end of the ACL.

Tip Enter the *acl_name* in upper case letters so the name is easy to see in the configuration. You might want to name the ACL for the interface (for example, INSIDE), or for the purpose (for example, NO_NAT or VPN).

Note You specify a network mask in the **access-list** command (for example, 255.255.255.0 for a class C mask). This method is different from the Cisco IOS software **access-list** command, which uses wildcard bits (for example, 0.0.0.255).

- Add an ACE for a specific protocol by entering the following command:

```
FWSM/contexta(config)# access-list acl_name [extended] {deny | permit} protocol
source_address mask dest_address mask
```

This type of ACE lets you specify any protocol for the source and destination addresses, but not ports. Typically, you identify **ip** for the protocol, but other protocols are accepted.

Enter **host** before the IP address to specify a single address. In this case, do not enter a mask. Enter **any** instead of the address and mask to specify any address.

For a list of protocol names, see the "Protocols and Applications" section on page D-5.

For information about logging options that you can add to the end of the ACE, see the "Logging Extended Access Control List Activity" section on page 10-26.

See the following examples:

The following ACL allows all hosts (on the interface to which you apply the ACL) to go through the FWSM:

```
FWSM/contexta(config)# access-list ACL_IN extended permit ip any any
```

The following sample ACL prevents hosts on 192.168.1.0/24 from accessing the 209.165.201.0/27 network. All other addresses are permitted:

```
FWSM/contexta(config)# access-list ACL_IN extended deny tcp 192.168.1.0 255.255.255.0
209.165.201.0 255.255.255.224
FWSM/contexta(config)# access-list ACL_IN extended permit ip any any
```

If you want to restrict access to only some hosts, then enter a limited permit ACE. By default, all other traffic is denied unless explicitly permitted.

```
FWSM/contexta(config)# access-list ACL_IN extended permit ip 192.168.1.0 255.255.255.0
209.165.201.0 255.255.255.224
```

- Add an ACE for TCP or UDP ports by entering the following command:

  ```
  FWSM/contexta(config)# access-list acl_name [extended] {deny | permit} {tcp | udp}
  source_address mask [operator port] dest_address mask [operator port]
  ```

  Enter **host** before the IP address to specify a single address. In this case, do not enter a mask. Enter **any** instead of the address and mask to specify any address.

  Use an *operator* to match port numbers used by the source or destination. The permitted operators are as follows:

  - **lt**—less than
  - **gt**—greater than
  - **eq**—equal to
  - **neq**—not equal to
  - **range**—an inclusive range of values. When you use this operator, specify two port numbers, for example:

    ```
    range 100 200
    ```

  For a list of permitted keywords and well-known port assignments, see the "TCP and UDP Ports" section on page D-6. DNS, Discard, Echo, Ident, NTP, RPC, SUNRPC, and Talk each require one definition for TCP and one for UDP. TACACS+ requires one definition for port 49 on TCP.

  For information about logging options that you can add to the end of the ACE, see the "Logging Extended Access Control List Activity" section on page 10-26.

  See the following example:

  The following ACL restricts all hosts (on the interface to which you apply the ACL) from accessing a website at address 209.165.201.29. All other traffic is allowed.

  ```
  FWSM/contexta(config)# access-list ACL_IN extended deny tcp any host 209.165.201.29 eq
  www
  FWSM/contexta(config)# access-list ACL_IN extended permit ip any any
  ```

- Add an ACE for ICMP by entering the following command:

```
FWSM/contexta(config)# access-list acl_name [extended] {deny | permit} icmp
source_address mask dest_address mask [icmp_type]
```

Enter **host** before the IP address to specify a single address. In this case, do not enter a mask. Enter **any** instead of the address and mask to specify any address.

Because ICMP is a connectionless protocol, you either need ACLs to allow ICMP in both directions (by applying ACLs to the source and destination interfaces), or you need to enable the ICMP inspection engine (see the "ICMP Inspection Engine" section on page 13-10). The ICMP inspection engine treats ICMP sessions as stateful connections.

To control ping, specify **echo-reply** (**0**) (FWSM to host) or **echo** (**8**) (host to FWSM). See the "ICMP Types" section on page D-9 for a list of ICMP types.

For information about logging options that you can add to the end of the ACE, see the "Logging Extended Access Control List Activity" section on page 10-26.

Step 2 To apply an extended ACL to the inbound or outbound direction of an interface, enter the following command:

```
FWSM/contexta(config)# access-group acl_name {in | out} interface interface_name
```

You can apply one ACL of each type (extended and EtherType) to both directions of the interface. See the "Inbound and Outbound Access Control Lists" section on page 10-10 for more information about ACL directions.

For connectionless protocols, you need to apply the ACL to the source and destination interfaces if you want traffic to pass in both directions. For example, you can allow BGP in an ACL in transparent mode, and you need to apply the ACL to both interfaces.

The following example illustrates the commands required to enable access to an inside web server with the IP address 209.165.201.12 (this IP address is the address visible on the outside interface after NAT):

```
FWSM/contexta(config)# access-list ACL_OUT extended permit tcp any host 209.165.201.12 eq
www
FWSM/contexta(config)# access-group ACL_OUT in interface outside
```

You also need to configure NAT for the web server. See the "Using Static NAT" section on page 9-25 for more information.

The following ACLs allow all hosts to communicate between the inside and hr networks, but only specific hosts to access the outside network:

```
FWSM/contexta(config)# access-list ANY extended permit ip any any
FWSM/contexta(config)# access-list OUT extended permit ip host 209.168.200.3 any
FWSM/contexta(config)# access-list OUT extended permit ip host 209.168.200.4 any

FWSM/contexta(config)# access-group ANY in interface inside
FWSM/contexta(config)# access-group ANY in interface hr
FWSM/contexta(config)# access-group OUT out interface outside
```

Adding an EtherType Access Control List

Transparent firewall mode only

An EtherType ACE controls any EtherType identified by a 16-bit hexadecimal number. You can identify some types by a keyword for convenience.

Because EtherTypes are connectionless, you need to apply the ACL to both interfaces if you want traffic to pass in both directions.

For example, you can permit or deny bridge protocol data units (BPDUs). By default, all BPDUs are denied. The FWSM receives trunk port (Cisco proprietary) BPDUs because FWSM ports are trunk ports. Trunk BPDUs have VLAN information inside the payload, so the FWSM modifies the payload with the outgoing VLAN if you allow BPDUs. If you use failover, you must allow BPDUs on both interfaces with an EtherType ACL to avoid bridging loops.

If you allow MPLS, ensure that Label Distribution Protocol (LDP) and Tag Distribution Protocol (TDP) TCP connections are established through the FWSM by configuring both MPLS routers connected to the FWSM to use the IP address on the FWSM interface as the router-id for LDP or TDP sessions. (LDP and TDP allow MPLS routers to negotiate the labels (addresses) used to forward packets.)

On Cisco IOS routers, enter the appropriate command for your protocol, LDP or TDP. The *interface* is the interface connected to the FWSM:

```
router(config)# mpls ldp router-id interface force
```

Or

```
router(config)# tag-switching tdp router-id interface force
```

You can apply only one ACL of each type (extended and EtherType) to each direction of an interface. You can also apply the same ACLs on multiple interfaces.

To add an EtherType ACL and apply it to an interface, follow these steps:

Step 1 Add one or more ACEs using the same ACL name by entering the following command:

```
FWSM/contexta(config)# access-list acl_name ethertype {permit | deny} {ipx | bpdu |
mpls-unicast | mpls-multicast | any | hex_number}
```

The *hex_number* is any EtherType that can be identified by a 16-bit hexadecimal number greater than or equal to 0x600. See RFC 1700, "Assigned Numbers," at http://www.ietf.org/rfc/rfc1700.txt for a list of EtherTypes.

When you enter the **access-list** command for a given ACL name, the ACE is added to the end of the ACL.

Tip Enter the *acl_name* in upper case letters so the name is easy to see in the configuration. You might want to name the ACL for the interface (for example, INSIDE), or for the purpose (for example, MPLS or IPX).

Step 2 To apply an EtherType ACL to the inbound or outbound direction of an interface, enter the following command:

```
FWSM/contexta(config)# access-group acl_name {in | out} interface interface_name
```

You can apply one ACL of each type (extended and EtherType) to both directions of the interface. See the "Inbound and Outbound Access Control Lists" section on page 10-10 for more information about ACL directions.

Because EtherTypes are connectionless, you need to apply the ACL to both interfaces if you want traffic to pass in both directions.

For example, the following sample ACL allows common EtherTypes originating on the inside interface:

```
FWSM/contexta(config)# access-list ETHER ethertype permit ipx
FWSM/contexta(config)# access-list ETHER ethertype permit bpdu
FWSM/contexta(config)# access-list ETHER ethertype permit mpls-unicast
FWSM/contexta(config)# access-group ETHER in interface inside
```

The following ACL allows some EtherTypes through the FWSM, but denies IPX:

```
FWSM/contexta(config)# access-list ETHER ethertype deny ipx
FWSM/contexta(config)# access-list ETHER ethertype permit 0x1234
FWSM/contexta(config)# access-list ETHER ethertype permit bpdu
FWSM/contexta(config)# access-list ETHER ethertype permit mpls-unicast
FWSM/contexta(config)# access-group ETHER in interface inside
FWSM/contexta(config)# access-group ETHER in interface outside
```

The following ACL denies traffic with EtherType 0x1256 but allows all others on both interfaces:

```
FWSM/contexta(config)# access-list nonIP ethertype deny 1256
FWSM/contexta(config)# access-list nonIP ethertype permit any
FWSM/contexta(config)# access-group ETHER in interface inside
FWSM/contexta(config)# access-group ETHER in interface outside
```

Adding a Standard Access Control List

Single context mode only

Standard ACLs identify the destination IP addresses of OSPF routes, and can be used in a route map for OSPF redistribution. Standard ACLs cannot be applied to interfaces to control traffic.

The following command adds a standard ACE. To add another ACE at the end of the ACL, enter another **access-list** command specifying the same ACL name. Apply the ACL using the "Adding a Route Map" section on page 8-6.

To add an ACE, enter the following command:

```
FWSM(config)# access-list acl_name standard {deny | permit} {any | ip_address mask}
```

The following sample ACL identifies routes to 192.168.1.0/24:

```
FWSM(config)# access-list OSPF standard permit 192.168.1.0 255.255.255.0
```

Simplifying Access Control Lists with Object Grouping

This section describes how to use object grouping to simplify ACL creation and maintenance, and includes the following topics:

How Object Grouping Works

By grouping like-objects together, you can use the object group in an ACE instead of having to enter an ACE for each object separately. You can create the following types of object groups:

- Protocol
- Network
- Service
- ICMP type

For example, consider the following three object groups:

- MyServices—Includes the TCP and UDP port numbers of the service requests that are allowed access to the internal network
- TrustedHosts—Includes the host and network addresses allowed access to the greatest range of services and servers
- PublicServers—Includes the host addresses of servers to which the greatest access is provided

After creating these groups, you could use a single ACE to allow trusted hosts to make specific service requests to a group of public servers.

You can also nest object groups in other object groups.

Note The ACE system limit applies to expanded ACLs. If you use object groups in ACEs, the number of actual ACEs that you enter is fewer, but the number of expanded ACEs is the same as without object groups. In many cases, object groups create more ACEs than if you added them manually, because creating ACEs manually leads you to summarize addresses more than an object group does. To view the number of expanded ACEs in an ACL, enter the **show access-list** *acl_name* command.

Adding Object Groups

This section describes how to add object groups, and includes the following topics:

- Adding a Protocol Object Group, page 10-19
- Adding a Network Object Group, page 10-19
- Adding a Service Object Group, page 10-20
- Adding an ICMP Type Object Group, page 10-21

Note If you add new members to an existing object group that is already in use by an ACE in a large ACL, recommitting the ACL can take a long time, depending on the size of the ACL and the object group. In some cases, making this change can cause the FWSM to devote over an hour to committing the ACL, during which time you cannot access the terminal. We recommend that you first remove the ACE that refers to the object group, make your change, and then add the ACE back to the ACL. See the "Manually Committing Access Control Lists and Rules" section on page 10-24 to insert an ACE in an ACL.

Adding a Protocol Object Group

To add or change a protocol object group, follow these steps. After you add the group, you can add more objects as required by following this procedure again for the same group name and specifying additional objects. You do not need to reenter existing objects; the commands you already set remain in place unless you remove them with the **no** form of the command.

To add a protocol group, follow these steps:

Step 1 To add a protocol group, enter the following command:

```
FWSM/contexta(config)# object-group protocol grp_id
```

The *grp_id* is a text string up to 64 characters in length.

The prompt changes to the protocol subcommand mode.

Step 2 (Optional) To add a description, enter the following command:

```
FWSM/contexta(config-protocol)# description text
```

The description can be up to 200 characters.

Step 3 To define the protocols in the group, enter the following command for each protocol:

```
FWSM/contexta(config-protocol)# protocol-object protocol
```

The *protocol* is the numeric identifier of the specific IP protocol (1 to 254) or a keyword identifier (for example, **icmp**, **tcp**, or **udp**). To include all IP protocols, use the keyword **ip**. For a list of protocols you can specify, see the "Protocols and Applications" section on page D-5.

For example, to create a protocol group for TCP, UDP, and ICMP, enter the following commands:

```
FWSM/contexta(config)# object-group protocol tcp_udp_icmp
FWSM/contexta(config-protocol)# protocol-object tcp
FWSM/contexta(config-protocol)# protocol-object udp
FWSM/contexta(config-protocol)# protocol-object icmp
```

Adding a Network Object Group

To add or change a network object group, follow these steps. After you add the group, you can add more objects as required by following this procedure again for the same group name and specifying additional objects. You do not need to reenter existing objects; the commands you already set remain in place unless you remove them with the **no** form of the command.

To add a network group, follow these steps:

Step 1 To add a network group, enter the following command:

```
FWSM/contexta(config)# object-group network grp_id
```

The *grp_id* is a text string up to 64 characters in length.

The prompt changes to the network subcommand mode.

Step 2 (Optional) To add a description, enter the following command:

```
FWSM/contexta(config-network)# description text
```

The description can be up to 200 characters.

Step 3 To define the networks in the group, enter the following command for each network or address:

```
FWSM/contexta(config-network)# network-object {host ip_address | ip_address mask}
```

For example, to create network group that includes the IP addresses of three administrators, enter the following commands:

```
FWSM/contexta(config)# object-group network admins
FWSM/contexta(config-network)# description Administrator Addresses
FWSM/contexta(config-network)# network-object host 10.1.1.4
FWSM/contexta(config-network)# network-object host 10.1.1.78
FWSM/contexta(config-network)# network-object host 10.1.1.34
```

Adding a Service Object Group

To add or change a service object group, follow these steps. After you add the group, you can add more objects as required by following this procedure again for the same group name and specifying additional objects. You do not need to reenter existing objects; the commands you already set remain in place unless you remove them with the **no** form of the command.

To add a service group, follow these steps:

Step 1 To add a service group, enter the following command:

```
FWSM/contexta(config)# object-group service grp_id {tcp | udp | tcp-udp}
```

The *grp_id* is a text string up to 64 characters in length.

Specify the protocol for the services (ports) you want to add, either **tcp**, **udp**, or **tcp-udp**. Enter **tcp-udp** if your service uses both TCP and UDP with the same port number, for example, DNS (port 53).

The prompt changes to the service subcommand mode.

Step 2 (Optional) To add a description, enter the following command:

```
FWSM/contexta(config-service)# description text
```

The description can be up to 200 characters.

Step 3 To define the ports in the group, enter the following command for each port or range of ports:

```
FWSM/contexta(config-service)# port-object {eq port | range begin_port end_port}
```

For a list of permitted keywords and well-known port assignments, see the "Protocols and Applications" section on page D-5.

For example, to create service groups that include DNS (TCP/UDP), LDAP (TCP), and RADIUS (UDP), enter the following commands:

```
FWSM/contexta(config)# object-group service services1 tcp-udp
FWSM/contexta(config-service)# description DNS Group
FWSM/contexta(config-service)# port-object eq domain
```

```
FWSM/contexta(config-service)# object-group service services2 udp
FWSM/contexta(config-service)# description RADIUS Group
FWSM/contexta(config-service)# port-object eq radius
FWSM/contexta(config-service)# port-object eq radius-acct

FWSM/contexta(config-service)# object-group service services3 tcp
FWSM/contexta(config-service)# description LDAP Group
FWSM/contexta(config-service)# port-object eq ldap
```

Adding an ICMP Type Object Group

To add or change an ICMP type object group, follow these steps. After you add the group, you can add more objects as required by following this procedure again for the same group name and specifying additional objects. You do not need to reenter existing objects; the commands you already set remain in place unless you remove them with the **no** form of the command.

To add an ICMP type group, follow these steps:

Step 1 To add an ICMP type group, enter the following command:

```
FWSM/contexta(config)# object-group icmp-type grp_id
```

The *grp_id* is a text string up to 64 characters in length.

The prompt changes to the ICMP type subcommand mode.

Step 2 (Optional) To add a description, enter the following command:

```
FWSM/contexta(config-icmp-type)# description text
```

The description can be up to 200 characters.

Step 3 To define the ICMP types in the group, enter the following command for each type:

```
FWSM/contexta(config-icmp-type)# icmp-object icmp_type
```

See the "ICMP Types" section on page D-9 for a list of ICMP types.

For example, to create an ICMP type group that includes echo-reply and echo (for controlling ping), enter the following commands:

```
FWSM/contexta(config)# object-group icmp-type ping
FWSM/contexta(config-service)# description Ping Group
FWSM/contexta(config-icmp-type)# icmp-object echo
FWSM/contexta(config-icmp-type)# icmp-object echo-reply
```

Nesting Object Groups

To nest an object group within another object group of the same type, first create the group that you want to nest according to the "Adding Object Groups" section on page 10-18. Then follow these steps:

Step 1 To add or edit an object group under which you want to nest another object group, enter the following command:

```
FWSM/contexta(config)# object-group {{protocol | network | icmp-type} grp_id |
service grp_id {tcp | udp | tcp-udp}}
```

Step 2 To add the specified group under the object group you specified in step 1, enter the following command:

```
FWSM/contexta(config-group_type)# group-object grp_id
```

The nested group must be of the same type.

You can mix and match nested group objects and regular objects within an object group.

For example, you create network object groups for privileged users from various departments:

```
FWSM/contexta(config)# object-group network eng
FWSM/contexta(config-network)# network-object host 10.1.1.5
FWSM/contexta(config-network)# network-object host 10.1.1.9
FWSM/contexta(config-network)# network-object host 10.1.1.89

FWSM/contexta(config-network)# object-group network hr
FWSM/contexta(config-network)# network-object host 10.1.2.8
FWSM/contexta(config-network)# network-object host 10.1.2.12

FWSM/contexta(config-network)# object-group network finance
FWSM/contexta(config-network)# network-object host 10.1.4.89
FWSM/contexta(config-network)# network-object host 10.1.4.100
```

You then nest all three groups together as follows:

```
FWSM/contexta(config)# object-group network admin
FWSM/contexta(config-network)# group-object eng
FWSM/contexta(config-network)# group-object hr
FWSM/contexta(config-network)# group-object finance
```

You only need to specify the admin object group in your ACE as follows:

```
FWSM/contexta(config)# access-list ACL_IN extended permit ip object-group admin host
209.165.201.29
```

Using Object Groups with an Access Control List

To use object groups in an ACL, replace the normal protocol (*protocol*), network (*source_address mask*, etc.), service (*operator port*), or ICMP type (*icmp_type*) parameter with **object-group** *grp_id*.

For example, to use object groups for all available parameters in the **access-list** {**tcp** | **udp**} command, enter the following command:

```
FWSM(config)# access-list acl_name [extended] {deny | permit} {tcp | udp} object-group
nw_grp_id [object-group svc_grp_id] object-group nw_grp_id [object-group svc_grp_id]
[log [[level] [interval secs] | disable | default]]
```

You do not have to use object groups for all parameters; for example, you can use an object group for the source address, but identify the destination address with an address and mask.

The following normal ACL that does not use object groups restricts several hosts on the inside network from accessing several web servers. All other traffic is allowed.

```
FWSM/contexta(config)# access-list ACL_IN extended deny tcp host 10.1.1.4 host
209.165.201.29 eq www
FWSM/contexta(config)# access-list ACL_IN extended deny tcp host 10.1.1.78 host
209.165.201.29 eq www
FWSM/contexta(config)# access-list ACL_IN extended deny tcp host 10.1.1.89 host
209.165.201.29 eq www
FWSM/contexta(config)# access-list ACL_IN extended deny tcp host 10.1.1.4 host
209.165.201.16 eq www
FWSM/contexta(config)# access-list ACL_IN extended deny tcp host 10.1.1.78 host
209.165.201.16 eq www
FWSM/contexta(config)# access-list ACL_IN extended deny tcp host 10.1.1.89 host
209.165.201.16 eq www
FWSM/contexta(config)# access-list ACL_IN extended deny tcp host 10.1.1.4 host
209.165.201.78 eq www
FWSM/contexta(config)# access-list ACL_IN extended deny tcp host 10.1.1.78 host
209.165.201.78 eq www
FWSM/contexta(config)# access-list ACL_IN extended deny tcp host 10.1.1.89 host
209.165.201.78 eq www
FWSM/contexta(config)# access-list ACL_IN extended permit ip any any
FWSM/contexta(config)# access-group ACL_IN in interface inside
```

If you make two network object groups, one for the inside hosts, and one for the web servers, then the configuration can be simplified and can be easily modified to add more hosts:

```
FWSM/contexta(config)# object-group network denied
FWSM/contexta(config-network)# network-object host 10.1.1.4
FWSM/contexta(config-network)# network-object host 10.1.1.78
FWSM/contexta(config-network)# network-object host 10.1.1.89

FWSM/contexta(config-network)# object-group network web
FWSM/contexta(config-network)# network-object host 209.165.201.29
FWSM/contexta(config-network)# network-object host 209.165.201.16
FWSM/contexta(config-network)# network-object host 209.165.201.78

FWSM/contexta(config-network)# access-list ACL_IN extended deny tcp object-group denied
object-group web eq www
FWSM/contexta(config)# access-list ACL_IN extended permit ip any any
FWSM/contexta(config)# access-group ACL_IN in interface inside
```

Displaying Object Groups

To display a list of the currently configured object groups, enter the following command:

```
FWSM/contexta(config)# show object-group [protocol | network | service | icmp-type |
id grp_id]
```

If you enter the command without any parameters, the system displays all configured object groups.

The following example shows sample output from the **show object-group** command.

```
FWSM/contexta# show object-group
object-group network ftp_servers
  description: This is a group of FTP servers
  network-object host 209.165.201.3
  network-object host 209.165.201.4
object-group network TrustedHosts
  network-object host 209.165.201.1
  network-object 192.168.1.0 255.255.255.0
  group-object ftp_servers
```

Removing Object Groups

To remove an object group, enter one of the following commands.

Note You cannot remove an object group or make an object group empty if it is used in an ACL.

- To remove a specific object group, enter the following command:

  ```
  FWSM/contexta(config)# no object-group grp_id
  ```

- To remove all object groups of the specified type, enter the following command:

  ```
  FWSM/contexta(config)# clear object-group [protocol | network | services | icmp-type]
  ```

  If you do not enter a type, all object groups are removed.

Manually Committing Access Control Lists and Rules

By default, the FWSM automatically commits ACLs as you enter them; the FWSM waits a short period of time after you last entered an **access-list** command before committing the ACL. See the "Access Control List Commit" section on page 10-6 for more information about committing ACLs.

You might want to manually commit ACLs if you have one of the following situations:

- You are running scripts and want to make sure the ACL was committed in its entirety. With auto-commit, you might commit partial ACLs if you run into memory limitations or other errors in the middle of the ACL entry.
- You want to modify an ACL, such as inserting lines, but do not want to disrupt traffic. For example, with auto-commit, you cannot insert a line into an ACL. You have to create a new ACL (with the inserted line), and then change the ACL name that is assigned to the interface, causing a brief

disruption. With manual commit, you can remove the ACL (from the configuration; not from running), enter a modified ACL with the same name, and then commit the ACL. Because the ACL name is the same, you do not need to change the interface assignment, and there is no disruption of traffic.

- You want to add several ACEs to a large ACL at the command line, and do not want the ACL to commit before you finish making your additions. For example, If you enter a line at the end of a 40,000 line ACL, and you do not enter each additional line within a second of the last line, then the ACL will commit each time you enter a line. A large ACL can take several minutes to commit, and you do not want to wait for the ACL to commit before entering the next line.

If you enable manual commit, then you must remember to manually commit any changes you make to ACLs or other rules, whether the change is an addition or a subtraction. Also, you must manually commit an ACL before you assign it to an interface (**access-group** command); the FWSM cannot assign an ACL to an interface if the ACL does not exist yet.

- To enable manual commit, or to return to auto-commit mode, enter the following command:

  ```
  FWSM/contexta(config)# access-list mode {manual-commit | auto-commit}
  ```

  Auto-commit is the default.

- To commit ACL changes in manual commit mode, enter the following command:

  ```
  FWSM/contexta(config)# access-list commit
  ```

- To view which ACLs are committed and which are uncommitted, enter the following command:

  ```
  FWSM/contexta(config)# show access-list
  ```

Adding Remarks to Access Control Lists

You can include remarks about entries in any ACL, including extended, EtherType, and standard ACLs. The remarks make the ACL easier to understand.

To add a remark after the last **access-list** command you entered, enter the following command:

```
FWSM/contexta(config)# access-list acl_id remark text
```

If you enter the remark before any **access-list** statements, then the remark is the first line in the ACL.

If you delete an ACL using the **no access-list** *acl_id* command, then all the remarks are also removed.

The text can be up to 100 characters in length. You can enter leading spaces at the beginning of the text. Trailing spaces are ignored.

For example, you can add remarks before each ACE, and the remark appears in the ACL in this location. Entering a dash (-) at the beginning of the remark helps set it apart from ACEs.

```
FWSM/contexta(config)# access-list OUT remark - this is the inside admin address
FWSM/contexta(config)# access-list OUT extended permit ip host 209.168.200.3 any
FWSM/contexta(config)# access-list OUT remark - this is the hr admin address
FWSM/contexta(config)# access-list OUT extended permit ip host 209.168.200.4 any
```

Logging Extended Access Control List Activity

This section describes how to configure ACL logging, and includes the following topics:

Access Control List Logging Overview

By default, when traffic is denied by an extended ACE, the FWSM generates system message 106023 for each denied packet, in the following form:

```
%FWSM-4-106023: Deny protocol src [interface_name:source_address/source_port] dst
interface_name:dest_address/dest_port [type {string}, code {code}] by access_group acl_id
```

If the FWSM is attacked, the number of system messages for denied packets can be very large. We recommend that you instead enable logging using system message 106100, which provides statistics for each ACE and lets you limit the number of system messages produced. Alternatively, you can disable all logging.

Note Only ACEs in the ACL generate logging messages; the implicit deny at the end of the ACL does not generate a message. If you want all denied traffic to generate messages, add the implicit ACE manually to the end of the ACL, as follows:

```
FWSM/contexta(config)# access-list TEST deny ip any any log
```

The **log** options at the end of the extended **access-list** command allow you to set the following behavior:

- Enable message 106100 instead of message 106023
- Disable all logging
- Return to the default logging using message 106023

System message 106100 is in the following form:

```
%FWSM-n-106100: access-list acl_id {permitted | denied} protocol
interface_name/source_address(source_port) -> interface_name/dest_address(dest_port)
hit-cnt number ({first hit | number-second interval})
```

When you enable logging for message 106100, if a packet matches an ACE, the FWSM creates a flow entry to track the number of packets received within a specific interval. The FWSM generates a system message at the first hit and at the end of each interval, identifying the total number of hits during the interval. At the end of each interval, the FWSM resets the hit count to 0. If no packets match the ACE during an interval, the FWSM deletes the flow entry.

A flow is defined by the source and destination IP addresses, protocols, and ports. Because the source port might differ for a new connection between the same two hosts, you might not see the same flow increment because a new flow was created for the connection. See the "Managing Deny Flows" section on page 10-28 to limit the number of logging flows.

Permitted packets that belong to established connections do not need to be checked against ACLs; only the initial packet is logged and included in the hit count. For connectionless protocols, such as ICMP, all packets are logged even if they are permitted, and all denied packets are logged.

See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module System Messages Guide* for detailed information about this system message.

Configuring Logging for an Access Control Entry

To configure logging for an ACE, see the following information about the **log** option:

```
FWSM/contexta(config)# access-list acl_name [extended] {deny | permit}...[log [[level]
[interval secs] | disable | default]]
```

See the "Adding an Extended Access Control List" section on page 10-13 for complete **access-list** syntax.

If you enter the **log** option without any arguments, you enable system log message 106100 at the default level (6) and for the default interval (300 seconds). See the following options:

- *level*—A severity level between 0 and 7. The default is 6.
- **interval** *secs*—The time interval in seconds between system messages, from 1 to 600. The default is 300. This value is also used as the timeout value for deleting an inactive flow.
- **disable**—Disables all ACL logging.

default—Enables logging to message 106023. This setting is the same as having no **log** option.

For example, you configure the following ACL:

```
FWSM/contexta(config)# access-list outside-acl permit ip host 1.1.1.1 any log 7 interval
600
FWSM/contexta(config)# access-list outside-acl permit ip host 2.2.2.2 any
FWSM/contexta(config)# access-list outside-acl deny ip any any log 2
FWSM/contexta(config)# access-group outside-acl in interface outside
```

When a packet is permitted by the first ACE of outside-acl, the FWSM generates the following system message:

```
%FWSM-7-106100: access-list outside-acl permitted tcp outside/1.1.1.1(12345) ->
inside/192.168.1.1(1357) hit-cnt 1 (first hit)
```

Although 20 additional packets for this connection arrive on the outside interface, the traffic does not have to be checked against the ACL, and the hit count does not increase.

If one more connection by the same host is initiated within the specified 10 minute interval (and the source and destination ports remain the same), then the hit count is incremented by 1 and the following message is displayed at the end of the 10 minute interval:

```
%FWSM-7-106100: access-list outside-acl permitted tcp outside/1.1.1.1(12345)->
inside/192.168.1.1(1357) hit-cnt 2 (600-second interval)
```

When a packet is denied by the third ACE, then the FWSM generates the following system message:

```
%FWSM-2-106100: access-list outside-acl denied ip outside/3.3.3.3(12345) ->
inside/192.168.1.1(1357) hit-cnt 1 (first hit)
```

20 additional attempts within a 5 minute interval (the default) result in the following message at the end of 5 minutes:

```
%FWSM-2-106100: access-list outside-acl denied ip outside/3.3.3.3(12345) ->
inside/192.168.1.1(1357) hit-cnt 21 (300-second interval)
```

Managing Deny Flows

When you enable logging for message 106100, if a packet matches an ACE, the FWSM creates a flow entry to track the number of packets received within a specific interval. The FWSM has a maximum of 32K logging flows for ACEs. A large number of flows can exist concurrently at any point of time. To prevent unlimited consumption of memory and CPU resources, the FWSM places a limit on the number of concurrent *deny* flows; the limit is placed only on deny flows (and not permit flows) because they can indicate an attack. When the limit is reached, the FWSM does not create a new deny flow for logging until the existing flows expire.

For example, if someone initiates a denial of service (DoS) attack, the FWSM can create a large number of deny flows in a short period of time. Restricting the number of deny flows prevents unlimited consumption of memory and CPU resources.

When you reach the maximum number of deny flows, the FWSM issues system message 106100:

```
%FWSM-1-106101: The number of ACL log deny-flows has reached limit (number).
```

To configure the maximum number of deny flows and to set the interval between deny flow alert messages (106101), enter the following commands:

- To set the maximum number of deny flows permitted per context before the FWSM stops logging, enter the following command:

  ```
  FWSM/contexta(config)# access-list deny-flow-max number
  ```

  The *number* is between 1 and 4096. 4096 is the default.

- To set the amount of time between system messages (number 106101) that identify that the maximum number of deny flows was reached, enter the following command:

  ```
  FWSM/contexta(config)# access-list alert-interval secs
  ```

  The *seconds* are between 1 and 3600. 300 is the default.

Allowing Remote Management

This chapter describes how to allow remote access to the Firewall Services Module (FWSM) CLI and how to allow ICMP to and from the FWSM.

Caution Management access to the FWSM using Telnet, SSH, or HTTPS might cause a degradation in performance depending on the commands that you execute during the session. For example, if there are 50,000 current connections and you enter the **show conn** command, the CPU utilization is higher than if you do not enter the command. We recommend that you avoid executing commands on the FWSM when high network performance is critical.

This chapter includes the following sections:

Note To "session" into the FWSM from the switch, see the "Sessioning and Logging into the Firewall Services Module" section on page 3-1.

Allowing Telnet

The FWSM allows Telnet connections to the FWSM for management purposes. You cannot use Telnet to the lowest security interface unless you use Telnet inside an IPSec tunnel (See the "Allowing a VPN Management Connection" section on page 11-5).

You can control the number of Telnet sessions allowed per context using resource classes (see the "Configuring a Class" section on page 5-14). The FWSM allows a maximum of 5 concurrent Telnet connections per context, if available, with a maximum of 100 connections divided between all contexts. See the "Rule Limits" section on page A-5 for information about the maximum number of Telnet rules allowed for the entire system.

To configure Telnet access to the FWSM, follow these steps:

Step 1 To identify the IP addresses from which the FWSM accepts connections, enter the following command for each address or subnet:

```
FWSM/contexta(config)# telnet source_IP_address mask source_interface
```

The *source_interface* cannot be the lowest security interface unless you use Telnet inside an IPSec tunnel (See the "Allowing a VPN Management Connection" section on page 11-5).

For example, you must configure at least two interfaces so the FWSM can determine the lowest security interface. If you configure a single interface (for the admin context, for example), then that interface is both the highest and the lowest security interface and cannot be used. Similarly, if all interfaces are on the same security level, you cannot use Telnet.

Step 2 (Optional) To set the duration for how long a Telnet session can be idle before the FWSM disconnects the session, enter the following command:

```
FWSM/contexta(config)# telnet timeout minutes
```

Set the timeout from 1 to 1440 minutes. The default is 5 minutes. The default duration is too short in most cases and should be increased until all pre-production testing and troubleshooting has been completed.

For example, to let a host on the inside interface with an address of 192.168.1.2 access the FWSM, enter the following command:

```
FWSM/contexta(config)# telnet 192.168.1.2 255.255.255.255 inside
FWSM/contexta(config)# telnet timeout 30
```

To allow all users on the 192.168.3.0 network to access the FWSM on the inside interface, enter the following command:

```
FWSM/contexta(config)# telnet 192.168.3.0 255.255.255.0 inside
```

Allowing SSH

The FWSM allows SSH connections to the FWSM for management purposes. You can control the number of SSH sessions allowed per context using resource classes (see the "Configuring a Class" section on page 5-14). The FWSM allows a maximum of 5 concurrent SSH connections per context, if available, with a maximum of 100 connections divided between all contexts. See the "Rule Limits" section on page A-5 for information about the maximum number of SSH rules allowed for the entire system.

SSH is an application running on top of a reliable transport layer, such as TCP/IP, that provides strong authentication and encryption capabilities. FWSM supports the SSH remote shell functionality provided in SSH Version 1 and supports DES and 3DES ciphers.

Note SSH v1.*x* and v2 are entirely different protocols and are not compatible. Make sure that you download a client that supports SSH v1.*x*.

This section includes the following topics:

- Configuring SSH Access, page 11-3
- Using an SSH Client, page 11-3

Configuring SSH Access

To configure SSH access to the FWSM, follow these steps:

Step 1 To generate an RSA key pair, which is required for SSH, enter the following command:

```
FWSM/contexta(config)# ca generate rsa key modulus
```

The modulus (in bits) is 512, 768, 1024, or 2048. The larger the key modulus size you specify, the longer it takes to generate an RSA. We recommend a value of 768.

Before you generate the key, you should set the host name and the domain name according to the "Setting the Host Name" section on page 6-4 and the "Setting the Domain Name" section on page 6-5. These settings are used in the key.

Step 2 To save the RSA keys to persistent Flash memory, enter the following command:

```
FWSM/contexta(config)# ca save all
```

Step 3 To identify the IP addresses from which the FWSM accepts connections, enter the following command for each address or subnet:

```
FWSM/contexta(config)# ssh source_IP_address mask source_interface
```

The FWSM accepts SSH connections from all interfaces, including the lowest security one.

Step 4 (Optional) To set the duration for how long an SSH session can be idle before the FWSM disconnects the session, enter the following command:

```
FWSM/contexta(config)# ssh timeout minutes
```

Set the timeout from 1 to 60 minutes. The default is 5 minutes. The default duration is too short in most cases and should be increased until all pre-production testing and troubleshooting has been completed.

For example, to generate RSA keys and let a host on the inside interface with an address of 192.168.1.2 access the FWSM, enter the following command:

```
FWSM/contexta(config)# ca generate rsa key 1024
FWSM/contexta(config)# ca save all
FWSM/contexta(config)# ssh 192.168.1.2 255.255.255.255 inside
FWSM/contexta(config)# ssh 192.168.1.2 255.255.255.255 inside
FWSM/contexta(config)# ssh timeout 30
```

To allow all users on the 192.168.3.0 network to access the FWSM on the inside interface, the following command:

```
FWSM/contexta(config)# ssh 192.168.3.0 255.255.255.0 inside
```

Using an SSH Client

To gain access to the FWSM console using SSH, at the SSH client enter the username **pix** and enter the login password set by the **password** command (see the "Changing the Login Password" section on page 6-2). For individual logins, see the "Configuring Authentication for CLI Access" section on page 12-8.

When starting an SSH session, a dot (.) displays on the FWSM console before the SSH user authentication prompt appears, as follows:

```
FWSM/contexta(config)# .
```

The display of the dot does not affect the functionality of SSH. The dot appears at the console when generating a server key or decrypting a message using private keys during SSH key exchange before user authentication occurs. These tasks can take up to two minutes or longer. The dot is a progress indicator that verifies that the FWSM is busy and has not hung.

Allowing HTTPS for PDM

To use PDM, you need to enable the HTTPS server, allow HTTPS connections to the FWSM, and enable the PDM metrics history. All of these tasks are completed if you use the **setup** command. This section describes how to manually configure PDM access.

The FWSM allows a maximum of 5 concurrent HTTPS connections per context, if available, with a maximum of 16 connections divided between all contexts. See the "Rule Limits" section on page A-5 for information about the maximum number of HTTPS rules allowed for the entire system.

To configure PDM access, follow these steps:

Step 1 To generate an RSA key pair, which is required for HTTPS, enter the following command:

```
FWSM/contexta(config)# ca generate rsa key modulus
```

The modulus (in bits) is 512, 768, 1024, or 2048. The larger the key modulus size you specify, the longer it takes to generate an RSA. We recommend a value of 768.

Before you generate the key, you should set the host name and the domain name according to the "Setting the Host Name" section on page 6-4 and the "Setting the Domain Name" section on page 6-5. These settings are used in the key.

Step 2 To save the RSA keys to persistent Flash memory, enter the following command:

```
FWSM/contexta(config)# ca save all
```

Step 3 To identify the IP addresses from which the FWSM accepts HTTPS connections, enter the following command for each address or subnet:

```
FWSM/contexta(config)# http source_IP_address mask source_interface
```

Step 4 To enable the HTTPS server, enter the following command:

```
FWSM/contexta(config)# http server enable
```

Step 5 To enable PDM metrics history, enter the following command:

```
FWSM/contexta(config)# pdm history enable
```

For example, to enable the HTTPS server and let a host on the inside interface with an address of 192.168.1.2 access PDM, enter the following commands:

```
FWSM/contexta(config)# ca generate rsa key 1024
FWSM/contexta(config)# ca save all
FWSM/contexta(config)# http server enable
FWSM/contexta(config)# pdm history enable
FWSM/contexta(config)# http 192.168.1.2 255.255.255.255 inside
```

To allow all users on the 192.168.3.0 network to access PDM on the inside interface, enter the following command:

```
FWSM/contexta(config)# http 192.168.3.0 255.255.255.0 inside
```

Allowing a VPN Management Connection

The FWSM supports IPSec for management access. An IPSec virtual private network (VPN) ensures that IP packets can safely travel over insecure networks such as the Internet. All communication between two VPN peers occurs over a secure tunnel, which means the packets are encrypted and authenticated by the peers.

The FWSM can connect to another VPN concentrator, such as a Cisco PIX firewall or a Cisco IOS router, using a site-to-site tunnel. You specify the peer networks that can communicate over the tunnel. In the case of the FWSM, the only address available on the FWSM end of the tunnel is the interface itself.

The FWSM can also accept connections from VPN clients, either hosts running the Cisco VPN client, or VPN concentrators such as the Cisco PIX firewall or Cisco IOS router running the Easy VPN client. Unlike a site-to-site tunnel, you do not know in advance the IP address of the client. Instead, you rely on client authentication.

The FWSM can support 5 concurrent IPSec connections, with a maximum of 10 concurrent connections divided between all contexts. You can control the number of IPSec sessions allowed per context using resource classes (see the "Configuring a Class" section on page 5-14).

This section describes the following topics:

- Configuring Basic Settings for All Tunnels, page 11-5
- Configuring VPN Client Access, page 11-7
- Configuring a Site-to-Site Tunnel, page 11-8

Configuring Basic Settings for All Tunnels

The following steps are required for both VPN client access and for site-to-site tunnels, and include setting the Internet Key Exchange (IKE) policy (IKE is part of the Internet Security Association and Key Management Protocol (ISAKMP)) and the IPSec transforms:

Step 1 To set the IKE encryption algorithm, enter the following command:

```
FWSM/contexta(config)# isakmp policy priority encryption {des | 3des}
```

The **3des** keyword is more secure than **des**.

You can have multiple IKE policies. The FWSM tries each policy in order of the *priority* until the policy matches the peer policy. The *priority* can be an integer from 1 to 65,534, with 1 being the highest priority and 65,534 the lowest. Use this same priority number for the following **isakmp** commands.

Step 2 To set the Diffie-Hellman group used for key exchange, enter the following command:

```
FWSM/contexta(config)# isakmp policy priority group {1 | 2}
```

Group 1 is 768 bits, and Group 2 is 1024 bits (and therefore more secure).

Step 3 To set the authentication algorithm, enter the following command:

```
FWSM/contexta(config)# isakmp policy priority hash {md5 | sha}
```

The **sha** keyword is more secure than **md5**.

Step 4 To set the IKE authentication method as a shared key, enter the following command:

```
FWSM/contexta(config)# isakmp policy priority authentication pre-share
```

You can alternatively use certificates instead of a shared key by specifying the **rsa-sig** option. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information about this method.

Step 5 To enable IKE on the tunnel interface, enter the following command:

```
FWSM/contexta(config)# isakmp enable interface_name
```

Step 6 To set the authentication and encryption methods used for IPSec tunnels in a transform set, enter the following command:

```
FWSM/contexta(config)# crypto ipsec transform-set transform_name {[ah-md5-hmac |
ah-sha-hmac] | [esp-md5-hmac | esp-sha-hmac]} {esp-des | esp-3des}
```

You refer to this transform set when you configure the VPN client group or a site-to-site tunnel.

You can refer to up to 6 transform sets for the tunnel, and the sets are checked in order until the transforms match.

The authentication and encryption algorithms of this transform typically match the IKE policy (**isakmp policy** commands). For site-to-site tunnels, this transform must match the peer transform.

Typically, you need to specify one authentication option and one encryption option.

Authentication options include the following (from most secure to least secure):

- **ah-sha-hmac**
- **ah-md5-hmac**
- **esp-sha-hmac**
- **esp-md5-hmac**

Encryption options include the following (from most secure to least secure):

- **esp-3des**
- **esp-des**

Note **esp-null** (no encryption) is for testing purposes only.

Although you can specify authentication alone, or encryption alone, these methods are not secure. You can also specify two authentication options, but this method does not increase security and also slows down the FWSM because each packet is authenticated two times.

For example, to configure the IKE policy and the IPSec transform sets, enter the following commands:

```
FWSM/contexta(config)# isakmp policy 1 authentication pre-share
FWSM/contexta(config)# isakmp policy 1 encryption 3des
FWSM/contexta(config)# isakmp policy 1 group 2
FWSM/contexta(config)# isakmp policy 1 hash sha
FWSM/contexta(config)# isakmp enable outside
FWSM/contexta(config)# crypto ipsec transform-set vpn_client esp-3des esp-sha-hmac
FWSM/contexta(config)# crypto ipsec transform-set site_to_site esp-3des ah-sha-hmac
```

Configuring VPN Client Access

A host with an installed version of the Cisco VPN Client can connect to the FWSM for management purposes over a public network, such as the Internet.

To allow remote clients to connect to the FWSM for management access, first configure basic VPN settings (see "Configuring Basic Settings for All Tunnels"), and then follow these steps:

Step 1 To specify the transform sets (defined in the "Configuring Basic Settings for All Tunnels" section on page 11-5) allowed for client tunnels, enter the following command:

```
FWSM/contexta(config)# crypto dynamic-map dynamic_map_name priority set transform-set
transform_set1 [transform_set2] [...]
```

List multiple transform sets in order of priority (highest priority first).

This dynamic crypto map allows unknown IP addresses to connect to the FWSM.

The **dynamic-map** name is used in Step 2.

The *priority* specifies the order in which multiple commands are evaluated. If you have a command that specifies one set of transforms, and another that specifies others, then the priority number determines the command that is evaluated first.

Step 2 To assign the dynamic crypto map (from Step 1) to a static tunnel, enter the following command:

```
FWSM/contexta(config)# crypto map crypto_map_name priority ipsec-isakmp dynamic
dynamic_map_name
```

Step 3 To specify the interface at which you want the client tunnels to terminate, enter the following command:

```
FWSM/contexta(config)# crypto map crypto_map_name interface interface_name
```

You can apply only one **crypto map** name to an interface, so if you want to terminate both a site-to-site tunnel and VPN clients on the same interface, they need to share the same **crypto map** name.

Step 4 To specify the AAA server or the local user database that provides user authentication when a client connects to the FWSM, enter the following command:

```
FWSM/contexta(config)# crypto map crypto_map_name client authentication
{LOCAL | aaa_server_name [LOCAL]}
```

You must first configure the server name according to the "Identifying a AAA Server" section on page 12-6 or the local database according to the "Configuring the Local Database" section on page 12-6.

Step 5 To specify the range of addresses that VPN clients use on the FWSM enter the following command:

```
FWSM/contexta(config)# ip local pool pool_name ip_address[-ip_address]
```

All tunneled packets from the client use one of these addresses as the source address.

Step 6 To specify the traffic that is destined for the FWSM, so you can tunnel only that traffic according to the **vpngroup split-tunnel** command in Step 8, enter the following command:

```
FWSM/contexta(config)# access-list acl_name [extended] permit {protocol} host
fwsm_interface_address pool_addresses mask
```

This ACL identifies traffic from the local pool (see Step 5) destined for the FWSM interface. See the "Adding an Extended Access Control List" section on page 10-13 for more information about ACLs.

Step 7 To assign the VPN address pool to a VPN group, enter the following command:

```
FWSM/contexta(config)# vpngroup group_name address-pool pool_name
```

This group specifies VPN characteristics for connecting clients. When a client connects the FWSM, they need to enter the VPN group name as well as the VPN group password in Step 9.

Step 8 To specify that only traffic destined for the FWSM is tunneled, enter the following command:

```
FWSM/contexta(config)# vpngroup group_name split-tunnel acl_name
```

This command is required.

Step 9 To set the VPN group password, enter the following command:

```
FWSM/contexta(config)# vpngroup group_name password password
```

Step 10 To allow Telnet or SSH access, see the "Allowing Telnet" section on page 11-1 and the "Allowing SSH" section on page 11-2.

Specify the VPN pool addresses in the **telnet** and **ssh** commands.

For example, the following commands allow VPN clients to use Telnet on the outside interface (209.165.200.225). The user authentication is the local database, so users with the VPN group name and password, as well as the username "admin" and the password "passw0rd" can connect to the FWSM.

```
FWSM/contexta(config)# isakmp policy 1 authentication pre-share
FWSM/contexta(config)# isakmp policy 1 encryption 3des
FWSM/contexta(config)# isakmp policy 1 group 2
FWSM/contexta(config)# isakmp policy 1 hash sha
FWSM/contexta(config)# isakmp enable outside
FWSM/contexta(config)# username admin password passw0rd
FWSM/contexta(config)# crypto ipsec transform-set vpn esp-3des esp-sha-hmac
FWSM/contexta(config)# crypto dynamic-map vpn_client 1 set transform-set vpn
FWSM/contexta(config)# crypto map telnet_tunnel 1 ipsec-isakmp dynamic vpn_client
FWSM/contexta(config)# crypto map telnet_tunnel interface outside
FWSM/contexta(config)# crypto map telnet_tunnel client authentication LOCAL
FWSM/contexta(config)# ip local pool client_pool 10.1.1.1-10.1.1.2
FWSM/contexta(config)# access-list VPN_SPLIT extended permit ip host 209.165.200.225 host
10.1.1.1
FWSM/contexta(config)# access-list VPN_SPLIT extended permit ip host 209.165.200.225 host
10.1.1.2
FWSM/contexta(config)# vpngroup admin address-pool client_pool
FWSM/contexta(config)# vpngroup admin split-tunnel VPN_SPLIT
FWSM/contexta(config)# vpngroup admin password $ecure23
FWSM/contexta(config)# telnet 10.1.1.1 255.255.255.255 outside
FWSM/contexta(config)# telnet 10.1.1.2 255.255.255.255 outside
FWSM/contexta(config)# telnet timeout 30
```

Configuring a Site-to-Site Tunnel

To configure a site-to-site tunnel, first configure basic VPN settings (see "Configuring Basic Settings for All Tunnels"), and then follow these steps:

Step 1 To set the shared key used by both peers, enter the following command:

```
FWSM/contexta(config)# isakmp key keystring address peer-address
```

Step 2 To identify the traffic allowed to go over the tunnel, enter the following command:

```
FWSM/contexta(config)# access-list acl_name [extended] {deny | permit} {protocol} host
fwsm_interface_address dest_address mask
```

For the destination address, specify the addresses that are allowed to access the FWSM.

See the "Adding an Extended Access Control List" section on page 10-13 for more information about ACLs.

Step 3 To create an IPSec tunnel, enter the following command:

```
FWSM/contexta(config)# crypto map crypto_map_name priority ipsec-isakmp
```

All tunnel attributes are identified by the same **crypto map** name.

The *priority* specifies the order in which multiple commands are evaluated. If you have a command for this **crypto map** name that specifies **ipsec-isakmp**, and another that specifies **ipsec-isakmp dynamic** (for VPN client connections), then the priority number determines the command that is evaluated first.

Step 4 To assign the ACL from Step 2 to this tunnel, enter the following command:

```
FWSM/contexta(config)# crypto map crypto_map_name priority match address acl_name
```

Step 5 To specify the remote peer on which this tunnel terminates, enter the following command:

```
FWSM/contexta(config)# crypto map crypto_map_name priority set peer ip_address
```

Step 6 To specify the transform sets for this tunnel (defined in the "Configuring Basic Settings for All Tunnels" section on page 11-5), enter the following command:

```
FWSM/contexta(config)# crypto map crypto_map_name priority set transform-set
transform_set1 [transform_set2] [...]
```

List multiple transform sets in order of priority (highest priority first). You can specify up to six transform sets.

Step 7 To specify the interface at which you want this tunnel to terminate, enter the following command:

```
FWSM/contexta(config)# crypto map crypto_map_name interface interface_name
```

You can apply only one **crypto map** name to an interface, so if you want to terminate both a site-to-site tunnel and VPN clients on the same interface, they need to share the same **crypto map** name.

This command must be entered after all other **crypto map** commands. If you change any **crypto map** settings, remove this command with the **no** prefix, and reenter it.

Step 8 To allow Telnet or SSH access, see the "Allowing Telnet" section on page 11-1 and the "Allowing SSH" section on page 11-2.

For example, the following commands allow hosts connected to the peer router (209.165.202.129) to use Telnet on the outside interface (209.165.200.225).

```
FWSM/contexta(config)# isakmp policy 1 authentication pre-share
FWSM/contexta(config)# isakmp policy 1 encryption 3des
FWSM/contexta(config)# isakmp policy 1 group 2
FWSM/contexta(config)# isakmp policy 1 hash sha
FWSM/contexta(config)# isakmp enable outside
FWSM/contexta(config)# crypto ipsec transform-set vpn esp-3des esp-sha-hmac
FWSM/contexta(config)# isakmp key 7mfi02lirotn address 209.165.200.223
FWSM/contexta(config)# access-list TUNNEL extended permit ip host 209.165.200.225
209.165.201.0 255.255.255.224
FWSM/contexta(config)# crypto map telnet_tunnel 2 ipsec-isakmp
FWSM/contexta(config)# crypto map telnet_tunnel 1 match address TUNNEL
```

```
FWSM/contexta(config)# crypto map telnet_tunnel 1 set peer 209.165.202.129
FWSM/contexta(config)# crypto map telnet_tunnel 1 set transform-set vpn
FWSM/contexta(config)# crypto map telnet_tunnel interface outside
FWSM/contexta(config)# telnet 209.165.201.0 255.255.255.224 outside
FWSM/contexta(config)# telnet timeout 30
```

Allowing ICMP to and from the FWSM

By default, ICMP (including ping) is not allowed to an FWSM interface (or through the FWSM. To allow ICMP *through* the FWSM, see Chapter 10, "Controlling Network Access with Access Control Lists."). ICMP is an important tool for testing your network connectivity; however, it can also be used to attack the FWSM or your network. We recommend allowing ICMP during your initial testing, but then disallowing it during normal operation.

See the "Rule Limits" section on page A-5 for information about the maximum number of ICMP rules allowed for the entire system.

To permit or deny address(es) to reach an FWSM interface with ICMP (either from a host to the FWSM, or from the FWSM to a host, which requires the ICMP reply to be allowed back), enter the following command:

```
FWSM/contexta(config)# icmp {permit | deny} {host ip_address | ip_address mask | any}
[icmp_type] interface_name
```

If you do not specify an *icmp_type*, all types are identified. You can enter the number or the name. To control ping, specify **echo-reply** (**0**) (FWSM to host) or **echo** (**8**) (host to FWSM). See the "ICMP Types" section on page D-9 for a list of ICMP types.

Like ACLs, the FWSM matches a packet to each **icmp** statement in order. You should use specific statements first, and general statements later. There is an implicit deny at the end. For example, if you allow all addresses first, then deny a specific address after, then that address will be unintentionally allowed because it matched the first statement.

Note If you only want to allow the FWSM to ping a host (and thus allow the echo reply back to the interface), and not allow hosts to ping the FWSM, you can enable the ICMP inspection engine instead of entering the command above. See the "ICMP Inspection Engine" section on page 13-10.

For example, to allow all hosts except the one at 10.1.1.15 to use ICMP to the inside interface, enter the following commands:

```
FWSM/contexta(config)# icmp deny host 10.1.1.15 inside
FWSM/contexta(config)# icmp permit any inside
```

To allow the host at 10.1.1.15 to use only ping to the inside interface, enter the following commands:

```
FWSM/contexta(config)# icmp permit host 10.1.1.15 inside
```

CHAPTER 12

Configuring AAA

Authentication, authorization, and accounting (AAA) tell the Firewall Services Module (FWSM) who the user is, what the user can do, and what the user did. This chapter contains the following sections:

Note See the "Rule Limits" section on page A-5 for information about the maximum number of AAA rules that are allowed for the entire system.

AAA Overview

AAA provides an extra level of protection and control for user access than using ACLs alone. For example, you can create an ACL allowing all outside users to access Telnet on a server on the DMZ network. If you want only some users to access the server, and you do not know their IP addresses, you can enable AAA to allow only authenticated and/or authorized users to make it through the FWSM. (The Telnet server has its own authentication; the FWSM prevents unauthorized users from attempting to access the server.)

You can use authentication alone or with authorization and accounting. Authorization always requires a user to be authenticated first. You can use accounting alone, or with authentication and authorization.

This section includes the following topics:

AAA Performance

The FWSM uses "cut-through proxy" to significantly speed up performance compared to a traditional proxy server. The performance of a traditional proxy server suffers because it analyzes every packet at the application layer of the Open System Interconnection (OSI) model. The FWSM cut-through proxy challenges a user initially at the application layer and then authenticates against standard Remote Authentication Dial-In User Service (RADIUS), Terminal Access Controller Access Control System Plus (TACACS+), or a local database. After the FWSM checks the policy, the FWSM shifts the session flow, and all traffic flows directly and quickly between the two parties while maintaining session state information.

About Authentication

Authentication lets you control access by requiring a valid username and password. You can configure the FWSM to authenticate the following items:

- All administrative connections to the FWSM including the following sessions:
  - Telnet
  - SSH
  - PDM (using HTTPS)
  - VPN management access (see the "Configuring VPN Client Access" section on page 11-7 for more information about using AAA with VPN)
- The **enable** command
- Network access through the FWSM

A user at a given IP address only needs to authenticate one time for all rules and types, until the authentication session expires. (See the **timeout uauth** command in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for timeout values.) For example, if you configure the FWSM to authenticate Telnet and FTP, and a user first successfully authenticates for Telnet, then as long as the session exists, the user does not also have to authenticate for FTP. See the "Authentication Overview" section on page 12-20 for more information about authentication sessions.

About Authorization

Authorization lets you control access *per user* after you authenticate with a valid username and password. You can configure the FWSM to authorize the following items:

- Management commands
- Network access through the FWSM

Authorization lets you control which services and commands are available to an individual user. Authentication alone provides the same access to services for all authenticated users.

If you need the control that authorization provides, you can configure a broad authentication rule, and then have a detailed authorization configuration. For example, you authenticate inside users to access any server on the outside network and then limit the outside servers that a particular user can access using authorization.

The FWSM caches the first 16 authorization requests per user, so if the user accesses the same services during the current authentication session, the FWSM does not resend the request to the authorization server.

About Accounting

Accounting lets you keep track of traffic that passes through the FWSM. If you enable authentication for that traffic, you can account for traffic per user. If you do not authenticate the traffic, you can account for traffic per IP address. Accounting information includes when sessions start and stop, the AAA client messages and username, the number of bytes that pass through the FWSM for the session, the service used, and the duration of each session.

AAA Server and Local Database Support

The FWSM supports AAA servers and a local database that is stored on the FWSM. Each server type and local database provides different functionality (see Table 12-1).

Table 12-1 AAA Server and Local Database Support

Server/Database Type	Functionality	Description
RADIUS	User authentication for CLI access	When a user attempts to access the FWSM for Telnet, SSH, or HTTP, the FWSM consults the RADIUS server for the username and password.
	User authentication for the **enable** command	When a user attempts to access the **enable** command, the FWSM consults the RADIUS server for the username and password.
	User authentication for network access	When a user attempts to access networks through the FWSM, and the traffic matches an authentication statement, the FWSM consults the RADIUS server for the username and password.
	User authorization for network access using downloaded ACLs per user (dynamic ACLs)	This user authorization occurs automatically when you configure authentication, but you must configure the RADIUS server to support it. When the user authenticates on the FWSM, the RADIUS server sends a dynamic ACL to the FWSM. The user's access to a given service is either permitted or denied by the ACL. The FWSM deletes the ACL when the authentication session expires.
	User authorization for network access using a downloaded ACL name per user	This user authorization occurs implicitly when you configure authentication, but you must configure the RADIUS server to support it. When the user authenticates on the FWSM, the RADIUS server sends a name of an ACL that is already defined on the FWSM. The user's access to a given service is either permitted or denied by the ACL. You can specify the same ACL for multiple users.
	VPN client authentication	When you configure VPN management access using the VPN client, you can use a RADIUS server to authenticate the client. (See the "Configuring VPN Client Access" section on page 11-7 for more information.)
	Accounting for network access per user or IP address	You can configure the FWSM to send accounting information to the RADIUS server about any traffic that passes through the FWSM.

Table 12-1 AAA Server and Local Database Support (continued)

Server/Database Type	Functionality	Description
TACACS+	User authentication for CLI access	When a user attempts to access the FWSM for Telnet, SSH, or HTTP, the FWSM consults the TACACS+ server for the username and password.
	User authentication for the **enable** command	When a user attempts to access the **enable** command, the FWSM consults the TACACS+ server for the username and password.
	User authentication for network access	When a user attempts to access networks through the FWSM, and the traffic matches an authentication statement, the FWSM consults the TACACS+ server for the username and password.
	User authorization for network access	When a user matches an authorization statement on the FWSM after authenticating, the FWSM consults the TACACS+ server for the user's access privileges.
	User authorization for management commands.	On the TACACS+ server, configure the commands that a user or group can use after they authenticate for CLI access. Every command that a user enters at the CLI is checked with the TACACS+ server.
	VPN client authentication	When you configure VPN management access using the VPN client, you can use a TACACS+ server to authenticate the client. (See the "Configuring VPN Client Access" section on page 11-7 for more information.)
	Accounting for network access per user or IP address	You can configure the FWSM to send accounting information to the TACACS+ server about any traffic that passes through the FWSM.
Local database[1]	User authentication for CLI access	When a user attempts to access the FWSM for Telnet, SSH, or HTTP, the FWSM consults the local user database for the username and password.
	User authentication for the **enable** or **login** command	When a user attempts to access the **enable** or **login** command, the FWSM consults the local user database for the username and password. You do not need to configure **login** user authentication; it is on by default.
	User authorization for management commands.	When a user authenticates with the **enable** command (or logs in with the **login** command), the FWSM places that user in the privilege level defined by the local database. You can configure each command to belong to privilege level between 0 and 15 on the FWSM.
	VPN client authentication	When you configure VPN management access using the VPN client, you can use the local database to authenticate the client. (See the "Configuring VPN Client Access" section on page 11-7 for more information.)

1. The local database can act as a fallback method for each of these functions if the AAA server is unavailable.

Configuring the Local Database

This section describes how to manage users in the local database. You can use the local database for CLI access authentication, privileged mode authentication, command authorization, or for VPN client authentication for management access. You cannot use the local database for network access authentication or authorization. For multiple context mode, you can configure usernames in the system execution space to provide individual logins using the **login** command; however, you cannot configure any **aaa** commands in the system execution space.

Caution If you add users to the local database who can gain access to the CLI and whom you do not want to enter privileged mode, you should enable command authorization. (See the "Configuring Local Command Authorization" section on page 12-10.) Without command authorization, users can access privileged mode (and all commands) at the CLI using their own password if their privilege level is 2 or greater (2 is the default). Alternatively, you can use RADIUS or TACACS+ authentication so the user will not be able to use the **login** command, or you can set all local users to level 1 so you can control who can use the system enable password to access privileged mode.

To define a user account in the local database, enter the following command:

```
FWSM/contexta(config)# username username {nopassword | password password}
[privilege level]
```

Define the following parameters:

- *username*—A string from 4 to 15 characters long.
- *password*—A string from 3 to 16 characters long.
- *privilege level*—The privilege level that you want to assign to the new user account (from 0 to 15). The default is 2. This privilege level is used with command authorization.
- **nopassword**—Creates a user account with no password.

For example, the following command assigns a privilege level of 15 to the admin user account:

```
FWSM/contexta(config)# username admin password passw0rd privilege 15
```

The following command creates a user account with no password:

```
FWSM/contexta(config)# username john.doe nopassword
```

Identifying a AAA Server

If you want to use an external AAA server (RADIUS or TACACS+) for authentication, authorization, or accounting, you must first add one or more servers to a server group on the FWSM. You identify this server group name when you add AAA rules. Each server group consists of only one type of server, RADIUS or TACACS+. For multiple context mode, you can configure up to 4 servers in a maximum of 4 groups. In single mode, you can configure 16 servers in a maximum of 14 server groups.

The FWSM contacts the first server in the group. If that server is unavailable, the FWSM contacts the next server in the group, if configured. If all servers in the group are unavailable, the FWSM tries the local database if you configured it as a fallback method (management authentication and authorization only). If you do not have a fallback method, the FWSM continues to try the AAA servers.

To add a server to a group, follow these steps:

Step 1 To identify the server group name and the protocol, enter the following command:

```
FWSM/contexta(config)# aaa-server server_group protocol {radius | tacacs+}
```

Step 2 To identify the maximum number of requests to send to a AAA server in the group before trying the next server, enter the following command:

```
FWSM/contexta(config)# aaa-server server_group max-failed-attempts number
```

The *number* can be between 1 and 5 times. The default is 3.

If you configured a fallback method using the local database (for management access only; see the "Configuring Authentication for CLI Access" section on page 12-8, the "Configuring Authentication to Access Privileged Mode" section on page 12-8, and the "Configuring TACACS+ Command Authorization" section on page 12-13 to configure the fallback mechanism), and all the servers in the group fail to respond, then the group is considered to be unresponsive, and the fallback method is tried. The server group remains marked as unresponsive for a period of 10 minutes (by default) so that additional AAA requests within that period do not attempt to contact the server group, and the fallback method is used immediately. To change the unresponsive period from the default, see the **aaa-server deadtime** command below.

If you do not have a fallback method, the FWSM continues to retry the servers in the group.

Step 3 If you configured a fallback method, identify the amount of time the server group is marked as unresponsive after all communications attempts fail by entering the following command:

```
FWSM/contexta(config)# aaa-server server_group deadtime minutes
```

Step 4 To add a server to the group, enter the following command:

```
FWSM/contexta(config)# aaa-server server_group (interface_name) host server_ip [key]
[timeout seconds]
```

The *key* is a case-sensitive, alphanumeric keyword of up to 127 characters that is the same value as the key on the server. Spaces are not permitted in the key, but other special characters are permitted. The key is used between the FWSM and server for encrypting data between them.

For example, to add one TACACS+ group with one primary and one backup server, and one RADIUS group with a single server, enter the following commands:

```
FWSM/contexta(config)# aaa-server AuthInbound protocol tacacs+
FWSM/contexta(config)# aaa-server AuthInbound max-failed-attempts 2
FWSM/contexta(config)# aaa-server AuthInbound deadtime 20
FWSM/contexta(config)# aaa-server AuthInbound (inside) host 10.1.1.1 TheUauthKey
FWSM/contexta(config)# aaa-server AuthInbound (inside) host 10.1.1.2 TheUauthKey2
FWSM/contexta(config)# aaa-server AuthOutbound protocol radius
FWSM/contexta(config)# aaa-server AuthOutbound (inside) host 10.1.1.3
```

Configuring Authentication for CLI Access

If you enable CLI authentication, the FWSM prompts you for your username and password to log in. After you enter your information, you have access to unprivileged mode.

To enter privileged mode, enter the **enable** command or the **login** command (if you are using the local database only).

If you configure enable authentication (see the "Configuring Authentication to Access Privileged Mode" section on page 12-8), the FWSM prompts you for your username and password. If you do not configure enable authentication, enter the system enable password when you enter the **enable** command (set by the **enable password** command). However, if you do not use enable authentication, after you enter the **enable** command, you are no longer logged in as a particular user. To maintain your username, use enable authentication.

For authentication using the local database, you can use the **login** command, which maintains the username but requires no configuration to turn on authentication.

Note Before the FWSM can authenticate a Telnet, SSH, or HTTP user, you must first configure access to the FWSM using the **telnet**, **ssh**, and **http** commands. These commands identify the IP addresses that are allowed to communicate with the FWSM. See Chapter 11, "Allowing Remote Management." The only exception is when you session from the switch to the FWSM; this Telnet session is always allowed. However, you cannot authenticate the system session because the system configuration does not contain any **aaa** commands.

To authenticate users who access the CLI, enter the following command:

```
FWSM/contexta(config)# aaa authentication {telnet | ssh | http} console {LOCAL | server_group [LOCAL]}
```

The **http** keyword authenticates the PDM client that accesses the FWSM using HTTPS.

If you use a TACACS+ or RADIUS server group for authentication, you can configure the FWSM to use the local database as a fallback method if the AAA server is unavailable. Specify the server group name followed by **LOCAL** (**LOCAL** is case sensitive). We recommend that you use the same username and password in the local database as the AAA server because the FWSM prompt does not give any indication which method is being used.

You can alternatively use the local database as your main method of authentication (with no fallback) by entering **LOCAL** alone.

Configuring Authentication to Access Privileged Mode

You can configure the FWSM to authenticate users with a AAA server or the local database when they enter the **enable** command. Alternatively, users are automatically authenticated with the local database when they enter the **login** command, which also accesses privileged mode depending on the user level in the local database. See the following sections for information about these methods:

- Configuring Authentication for the enable Command, page 12-9
- Authenticating Users Using the login Command, page 12-9

Configuring Authentication for the enable Command

You can configure the FWSM to authenticate users when they enter the **enable** command. If you do not authenticate the **enable** command, when you enter **enable**, the FWSM prompts for the system enable password (set by the **enable password** command), and you are no longer logged in as a particular user. Enable authentication maintains the username. This feature is particularly useful when you perform command authorization, where usernames are important to determine the commands a user can enter.

To authenticate users who enter the **enable** command, enter the following command:

```
FWSM/contexta(config)# aaa authentication enable console {LOCAL | server_group [LOCAL]}
```

The user is prompted for the username and password.

If you use a TACACS+ or RADIUS server group for authentication, you can configure the FWSM to use the local database as a fallback method if the AAA server is unavailable. Specify the server group name followed by **LOCAL** (**LOCAL** is case sensitive). We recommend that you use the same username and password in the local database as the AAA server because the FWSM prompt does not give any indication which method is being used.

You can alternatively use the local database as your main method of authentication (with no fallback) by entering **LOCAL** alone.

Authenticating Users Using the login Command

From unprivileged mode, you can log in as any username in the local database using the **login** command. Unlike enable authentication, this method is available in the system execution space in multiple context mode.

This feature allows users to log in with their own username and password to access privileged mode, so you do not have to give out the system enable password to everyone. To allow users to access privileged mode (and all commands) when they log in, set the user privilege level to 2 (the default) through 15. If you configure local command authorization, then the user can only enter commands assigned to that privilege level or lower. See the "Configuring Local Command Authorization" section on page 12-10 for more information.

Caution If you add users to the local database who can gain access to the CLI and whom you do not want to enter privileged mode, you should configure command authorization. Without command authorization, users can access privileged mode (and all commands) at the CLI using their own password if their privilege level is 2 or greater (2 is the default). Alternatively, you can use RADIUS or TACACS+ authentication, or you can set all local users to level 1 so you can control who can use the system enable password to access privileged mode.

To log in as a user from the local database, enter the following command:

```
FWSM> login
```

The FWSM prompts for your username and password. After you enter your password, the FWSM places you in the privilege level that the local database specifies.

Configuring Command Authorization

By default when you log in, you can access unprivileged mode, which offers only minimal commands. When you enter the **enable** command (or the **login** command when you use the local database), you can access privileged mode and advanced commands, including configuration commands. If you want to control the access to commands, the FWSM lets you configure command authorization, where you can determine which commands that are available to a user.

This section includes the following topics:

Command Authorization Overview

You can use one of two command authorization methods:

- Local database—Configure the command privilege levels on the FWSM. When a local user authenticates with the **enable** command (or logs in with the **login** command), the FWSM places that user in the privilege level that is defined by the local database. The user can then access commands at the user's privilege level and below.

Note You can use local command authorization without any users in the local database and without CLI or enable authentication. Instead, when you enter the **enable** command, you enter the system enable password, and the FWSM places you in level 15. You can then create enable passwords for every level, so that when you enter **enable** *n* (2 to 15), the FWSM places you in level *n*. These levels are not used unless you turn on local command authorization (see "Configuring Local Command Authorization" below). (See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information about **enable**.)

- TACACS+ server—On the TACACS+ server, configure the commands that a user or group can use after they authenticate for CLI access. Every command that a user enters at the CLI is checked with the TACACS+ server.

Configuring Local Command Authorization

Local command authorization places each user at a privilege level, and each user can enter any command at their privilege level or below. The FWSM lets you assign commands to one of 16 privilege levels (0 to 15). By default, each command is assigned either to privilege level 0 or to privilege level 15.

This section includes the following topics:

Local Command Authorization Prerequisites

Complete the following tasks as part of your command authorization configuration:

- Configure **enable** authentication. (See the "Configuring Authentication to Access Privileged Mode" section on page 12-8.)

  Alternatively, you can use the **login** command (which is the same as the **enable** command with authentication), which requires no configuration. We do not recommend this option because it is not as secure as enable authentication.

  You can also use CLI authentication (see the "Configuring Authentication for CLI Access" section on page 12-8), but it is not required.

- Configure each user in the local database at a privilege level from 0 to 15. (See the "Configuring the Local Database" section on page 12-6.)

Default Command Privilege Levels

By default, the following commands are assigned to privilege level 0. All other commands are at level 15.

- **show checksum**
- **show curpriv**
- **enable** (enable mode)
- **help**
- **show history**
- **login**
- **logout**
- **pager**
- **show pager**
- **clear pager**
- **quit**
- **show version**

If you move any configure mode commands to a lower level than 15, be sure to move the **configure** command to that level as well, otherwise, the user will not be able to enter configuration mode.

To view all privilege levels, see the "Viewing Command Privilege Levels" section on page 12-13.

Assigning Privilege Levels to Commands and Enabling Authorization

To assign a command to a new privilege level, and enable authorization, follow these steps:

Step 1 To assign a command to a privilege level, enter the following command:

```
FWSM/contexta(config)# privilege [show | clear | configure] level level
[mode {enable | configure}] command command
```

Repeat this command for each command you want to reassign.

See the following information about the options in this command:

- **show | clear | configure**—These optional keywords allow you to set the privilege only for the show, clear, or configure form of the command. The configure form of the command is typically the form that causes a configuration change, either as the unmodified command (without the **show** or **clear** prefix) or as the **no** form. If you do not use one of these keywords, all forms of the command are affected.
- **level** *level*—A level between 0 and 15.
- **mode** {**enable | configure**}—If a command can be entered in unprivileged/privileged mode as well as configuration mode, and the command performs different actions in each mode, you can set the privilege level for these modes separately:
  - **enable**—Specifies both unprivileged mode and privileged mode.
  - **configure**—Specifies configuration mode, accessed using the **configure terminal** command.
- **command** *command*—The command you are configuring. You can only configure the privilege level of the *main* command. For example, you can configure the level of all **aaa** commands, but not the level of the **aaa authentication** command and the **aaa authorization** command separately.

  Also, you cannot configure the privilege level of subcommands separately from the main command. For example, you can configure the **context** command, but not the **allocate-interface** command, which inherits the settings from the **context** command.

Step 2 To enable local command authorization, enter the following command:

```
FWSM/contexta(config)# aaa authorization command LOCAL
```

Even if you set command privilege levels, command authorization does not take place unless you enable command authorization with this command.

For example, the **filter** command has the following forms:

- **filter** (represented by the **configure** option)
- **show filter**
- **clear filter**

You can set the privilege level separately for each form, or set the same privilege level for all forms by omitting this option. For example, set each form separately as follows:

```
FWSM/contexta(config)# privilege show level 5 command filter
FWSM/contexta(config)# privilege clear level 10 command filter
FWSM/contexta(config)# privilege configure level 10 command filter
```

Alternatively, you can set all filter commands to the same level:

```
FWSM/contexta(config)# privilege level 5 command filter
```

The **show privilege** command separates the forms in the display.

The following example shows the use of the **mode** keyword. The **enable** command must be entered from unprivileged mode, while the **enable password** command, which is accessible in configuration mode, requires the highest privilege level.

```
FWSM/contexta(config)# privilege configure level 0 mode enable command enable
FWSM/contexta(config)# privilege configure level 15 mode configure command enable
FWSM/contexta(config)# privilege show level 15 mode configure command enable
```

This example shows an additional command, the **configure** command, that uses the **mode** keyword:

```
FWSM/contexta(config)# privilege show level 5 mode configure command configure
FWSM/contexta(config)# privilege clear level 15 mode configure command configure
```

```
FWSM/contexta(config)# privilege configure level 15 mode configure command configure
FWSM/contexta(config)# privilege configure level 15 mode enable command configure
```

Note This last line is for the **configure terminal** command.

Viewing Command Privilege Levels

The following commands allow you to view privilege levels for commands.

- To show all commands, enter the following command:

  ```
  FWSM/contexta(config)# show privilege all
  ```

- To shows command for a specific level, enter the following command:

  ```
  FWSM/contexta(config)# show privilege level level
  ```

  The *level* is an integer between 0 and 15.

- To show the level of a specific command, enter the following command:

  ```
  FWSM/contexta(config)# show privilege command command
  ```

For example, for the **show privilege all** command, the system displays the current assignment of each CLI command to a privilege level. The following example illustrates the first part of the display:

```
FWSM(config)# show privilege all
privilege show level 15 command aaa
privilege clear level 15 command aaa
privilege configure level 15 command aaa
privilege show level 15 command aaa-server
privilege clear level 15 command aaa-server
privilege configure level 15 command aaa-server
privilege show level 15 command access-group
privilege clear level 15 command access-group
privilege configure level 15 command access-group
privilege show level 15 command access-list
privilege clear level 15 command access-list
privilege configure level 15 command access-list
privilege show level 15 command activation-key
privilege configure level 15 command activation-key
....
```

The following command displays the command assignments for privilege level 10:

```
FWSM/contexta(config)# show privilege level 10
privilege show level 10 command aaa
```

The following command displays the command assignment for the **access-list** command:

```
FWSM/contexta(config)# show privilege command access-list
privilege show level 15 command access-list
privilege clear level 15 command access-list
privilege configure level 15 command access-list
```

Configuring TACACS+ Command Authorization

If you enable TACACS+ command authorization, and a user enters a command at the CLI, the FWSM sends the command and username to the TACACS+ server to determine if the command is authorized.

When configuring command authorization with a TACACS+ server, do not save your configuration until you are sure it works the way you want. If you get locked out because of a mistake, you can usually recover access by restarting the FWSM. If you still get locked out, see the "Recovering from a Lockout" section on page 12-19.

Be sure that your TACACS+ system is completely stable and reliable. The necessary level of reliability typically requires that you have a fully redundant TACACS+ server system and fully redundant connectivity to the FWSM. For example, in your TACACS+ server pool, include one server connected to interface 1, and another to interface 2. You can also configure local command authorization as a fallback method if the TACACS+ server is unavailable. In this case, you need to configure local users and command privilege levels according to the "Configuring Local Command Authorization" section on page 12-10.

This section includes the following topics:

TACACS+ Command Authorization Prerequisites

Complete the following tasks as part of your command authorization configuration:

- Configure CLI authentication (see the "Configuring Authentication for CLI Access" section on page 12-8).
- Configure **enable** authentication (see the "Configuring Authentication to Access Privileged Mode" section on page 12-8).

Configuring Commands on the TACACS+ Server

You can configure commands on a CiscoSecure Access Control Server (ACS) TACACS+ server as a shared profile component, for a group, or for individual users. For third-party TACACS+ servers, see your server documentation for more information about command authorization support.

See the following guidelines for configuring commands on a CiscoSecure ACS TACACS+ server Version 3.1; many of these guidelines also apply to third-party servers:

- The FWSM sends the commands to be authorized as "shell" commands, so configure the commands on the TACACS+ server as shell commands.

Note The Cisco Secure ACS server might include a command type called "pix-shell." Do not use this type for FWSM command authorization.

- The first word of the command is considered to be the main command. All additional words are considered to be arguments, which need to be preceded by **permit** or **deny**.

  For example, to allow **show aaa**, **aaa authentication**, and **aaa authorization command** commands, add **aaa** to the command box, and type **permit authentication** and **permit authorization command** in the arguments box. The **show aaa** command must be listed separately (see Figure 12-1 and Figure 12-2).

Figure 12-1 Permitting Specific Commands

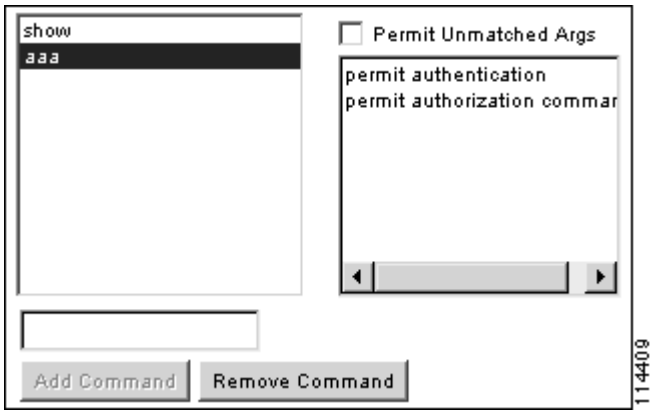
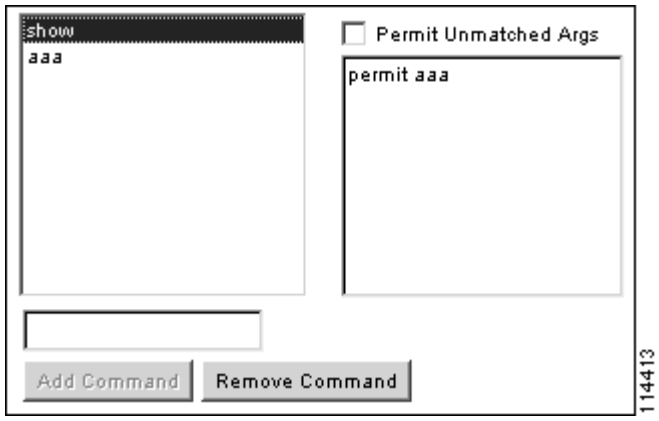


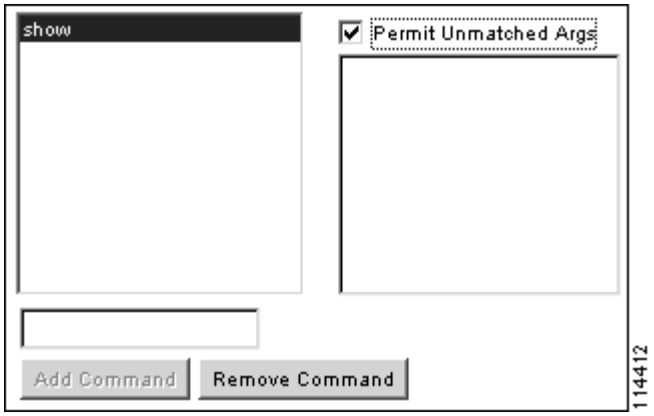
Figure 12-2 Permitting the Show Version of a Command



- You can permit all arguments of a command that you do not explicitly deny by selecting the Permit Unmatched Args check box.

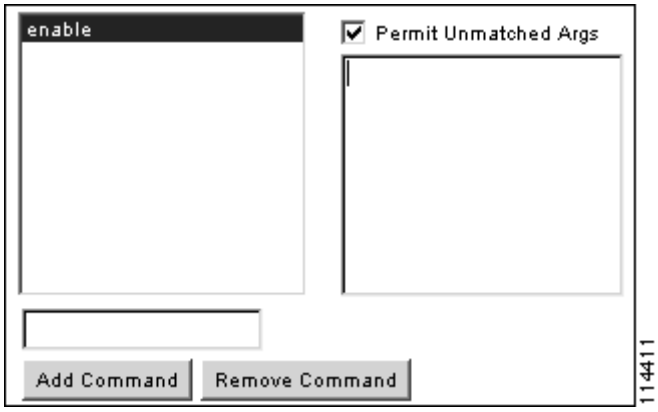
  For example, you can configure just the **show** command, and then all **show** commands are allowed. We recommend using this method so that you do not have to anticipate every variant of a command, including abbreviations and **?**, which shows CLI usage (see Figure 12-3).

Figure 12-3 Permitting All Related Commands



- For commands that are a single word, you *must* permit unmatched arguments, even if there are no arguments for the command, for example **enable** or **help** (see Figure 12-4).

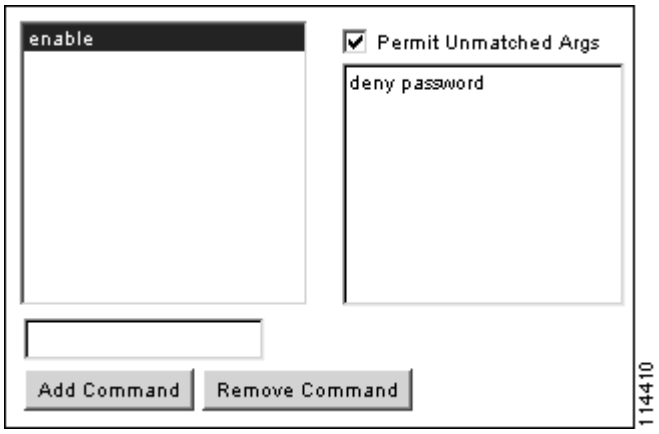
Figure 12-4 Permitting Single Word Commands



- To disallow some arguments, enter the arguments preceded by **deny**.

  For example, to allow **enable**, but not **enable password**, enter **enable** in the commands box, and **deny password** in the arguments box. Be sure to select the Permit Unmatched Args check box so that **enable** alone is still allowed (see Figure 12-5).

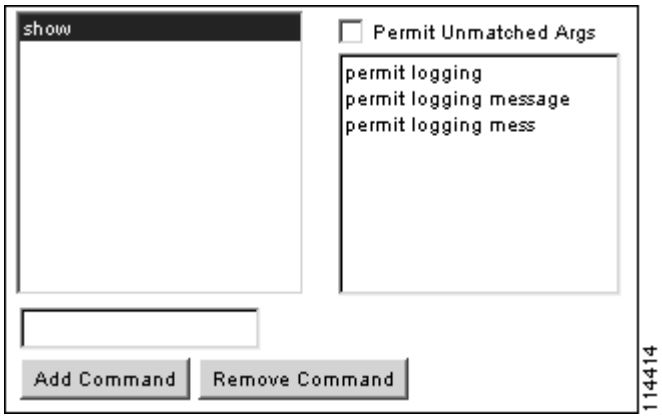
Figure 12-5 Disallowing Arguments



- When you abbreviate a command at the command line, the FWSM expands the prefix and main command to the full text, but it sends additional arguments to the TACACS+ server as you enter them.

  For example, if you enter **sh log**, then the FWSM sends the entire command to the TACACS+ server, **show logging**. However, if you enter **sh log mess**, then the FWSM sends **show logging mess** to the TACACS+ server, and not the expanded command **show logging message**. You can configure multiple spellings of the same argument to anticipate abbreviations (see Figure 12-6).

Figure 12-6 Specifying Abbreviations



- We recommend that you allow the following basic commands for all users:
  - **show checksum**
  - **show curpriv**
  - **enable**
  - **help**
  - **show history**
  - **login**
  - **logout**
  - **pager**
  - **show pager**
  - **clear pager**
  - **quit**
  - **show version**

Enabling TACACS+ Command Authorization

Before you enable TACACS+ command authorization, be sure that you are logged into the FWSM as a user that is defined on the TACACS+ server, and that you have the necessary command authorization to continue configuring the FWSM. For example, you should log in as an admin user with all commands authorized. Otherwise, you could become unintentionally locked out.

To perform command authorization using a TACACS+ server, enter the following command:

```
FWSM/contexta(config)# aaa authorization command tacacs+_server_group [LOCAL]
```

You can configure the FWSM to use the local database as a fallback method if the TACACS+ server is unavailable. To enable fallback, specify the server group name followed by **LOCAL** (**LOCAL** is case sensitive). We recommend that you use the same username and password in the local database as the TACACS+ server because the FWSM prompt does not give any indication which method is being used.

Be sure to configure users in the local database (see the "Configuring the Local Database" section on page 12-6) and command privilege levels (see the "Configuring Local Command Authorization" section on page 12-10).

Viewing the Current Logged-In User

To view the current logged-in user, enter the following command:

```
FWSM/contexta# show curpriv
```

See the following sample **show curpriv** command output. A description of each field follows.

```
FWSM/contexta# show curpriv
Username : admin
Current privilege level : 15
Current Mode/s : P_PRIV
```

Table 12-2 describes the **show curpriv** command output.

Table 12-2 show curpriv Display Description

Field	Description
Username	Username. If you are logged in as the default user, the name is enable_1 (unprivileged) or enable_15 (privileged).
Current privilege level	Level from 0 to 15. Unless you configure local command authorization and assign commands to intermediate privilege levels, levels 0 and 15 are the only levels that are used.
Current Mode/s	Shows the access modes: • P_UNPR—Unprivileged mode (levels 0 and 1) • P_PRIV—Privileged mode (levels 2 to 15) • P_CONF—Configuration mode

Recovering from a Lockout

In some circumstances, when you turn on command authorization or CLI authentication, you can be locked out of the FWSM CLI. You can usually recover access by restarting the FWSM. However, if you already saved your configuration, you might be locked out. Table 12-3 lists the common lockout conditions and how you might recover from them.

Table 12-3 CLI Authentication and Command Authorization Lockout Scenarios

Feature	Lockout Condition	Description	Workaround: Single Mode	Workaround: Multiple Mode
Local CLI authentication	No users in the local database	If you have no users in the local database, you cannot log in, and you cannot add any users.	Log into the maintenance partition and reset the passwords and **aaa** commands. See the "Clearing the Application Partition Passwords and AAA Settings" section on page 17-9.	Session into the FWSM from the switch. From the system execution space, you can change to the context and add a user.
TACACS+ command authorization TACACS+ CLI authentication RADIUS CLI authentication	Server down or unreachable and you do not have the fallback method configured	If the server is unreachable, then you cannot log in or enter any commands.	1. Log into the maintenance partition and reset the passwords and AAA commands. See the "Clearing the Application Partition Passwords and AAA Settings" section on page 17-9. 2. Configure the local database as a fallback method so you do not get locked out when the server is down.	1. If the server is unreachable because the network configuration is incorrect on the FWSM, session into the FWSM from the switch. From the system execution space, you can change to the context and reconfigure your network settings. 2. Configure the local database as a fallback method so you do not get locked out when the server is down.

Table 12-3 CLI Authentication and Command Authorization Lockout Scenarios (continued)

Feature	Lockout Condition	Description	Workaround: Single Mode	Workaround: Multiple Mode
TACACS+ command authorization	You are logged in as a user without enough privileges or as a user that does not exist	You enable command authorization, but then find that the user cannot enter any more commands.	Fix the TACACS+ server user account. If you do not have access to the TACACS+ server and you need to configure the FWSM immediately, then log into the maintenance partition and reset the passwords and **aaa** commands. See the "Clearing the Application Partition Passwords and AAA Settings" section on page 17-9.	Session into the FWSM from the switch. From the system execution space, you can change to the context and complete the configuration changes. You can also disable command authorization until you fix the TACACS+ configuration.
Local command authorization	You are logged in as a user without enough privileges	You enable command authorization, but then find that the user cannot enter any more commands.	Log into the maintenance partition and reset the passwords and **aaa** commands. See the "Clearing the Application Partition Passwords and AAA Settings" section on page 17-9.	Session into the FWSM from the switch. From the system execution space, you can change to the context and change the user level.

Configuring Authentication for Network Access

This section includes the following topics:

Authentication Overview

The FWSM lets you configure network access authentication using RADIUS or TACACS+ servers.

Although you can configure network access authentication for any protocol or service, you can authenticate directly with HTTP, Telnet, or FTP only. A user must first authenticate with one of these services before other traffic that requires authentication is allowed through. If you do not want to allow HTTP, Telnet, or FTP through the FWSM, but want to authenticate other types of traffic, you can configure virtual Telnet; the user Telnets to a given IP address configured on the FWSM, and the FWSM provides a Telnet prompt. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information about the **virtual telnet** command.

A user at a given IP address only needs to authenticate one time for all rules and types, until the authentication session expires. (See the **timeout uauth** command in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for timeout values.) For

example, if you configure the FWSM to authenticate Telnet and FTP, and a user first successfully authenticates for Telnet, then as long as the authentication session exists, the user does not also have to authenticate for FTP.

For Telnet, HTTP, and FTP, the FWSM generates an authentication prompt. If the destination server also has its own authentication, the user enters another username and password.

Note If you use HTTP authentication, the RADIUS or TACACS+ username and password are sent in clear text to the destination web server, and not just to the AAA server. Therefore, you should enable HTTP authentication with caution. For example, if you authenticate inside users when they access outside web servers, anyone on the outside can learn the user's RADIUS or TACACS+ username and password. We recommend that you use URL filtering if you want to control external web access. You can also use virtual HTTP, which allows the FWSM to authenticate HTTP users directly and then forward the requests to the final destination. This feature can have a serious impact on performance, however. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information about the **virtual http** command.

For FTP, a user has the option of entering the FWSM username followed by an at sign (@) and then the FTP username (name1@name2). For the password, the user enters the FWSM password followed by an at sign (@) and then the FTP password (password1@password2). For example, enter the following text:

```
name> john_c@jchrichton
password> letmein@he110
```

This feature is useful when you have cascaded firewalls that require multiple logins. You can separate several names and passwords by multiple at signs (@).

Enabling Network Access Authentication

To configure authentication, enter the following command:

```
FWSM/contexta(config)# aaa authentication match acl_name interface_name server_group
```

Identify the source addresses and destination addresses using an extended ACL. Create the ACL using the **access-list** command (see the "Adding an Extended Access Control List" section on page 10-13). The **permit** access control entries (ACEs) mark matching traffic for authentication, while **deny** entries exclude matching traffic from authentication. Be sure to include the destination ports for either HTTP, Telnet, or FTP in the ACL because the user must authenticate with one of these services before other services are allowed through the FWSM.

Note You can alternatively use the **aaa authentication include** command (which identifies traffic within the command). However, you cannot use both methods in the same configuration. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information.

For example, the following commands authenticate all inside HTTP traffic and SMTP traffic:

```
FWSM/contexta(config)# access-list MAIL_AUTH extended permit tcp any any eq smtp
FWSM/contexta(config)# access-list MAIL_AUTH extended permit tcp any any eq www
FWSM/contexta(config)# aaa-server AuthOutbound protocol tacacs+
FWSM/contexta(config)# aaa-server AuthOutbound (inside) host 10.1.1.1 TheUauthKey
```

```
FWSM/contexta(config)# aaa authentication match MAIL_AUTH inside AuthOutbound
```

The following commands authenticate Telnet traffic from the outside interface to a particular server (209.165.201.5):

```
FWSM/contexta(config)# access-list TELNET_AUTH extended permit tcp any host 209.165.201.5
eq telnet
FWSM/contexta(config)# aaa-server AuthInbound protocol tacacs+
FWSM/contexta(config)# aaa-server AuthInbound (inside) host 10.1.1.1 TheUauthKey
FWSM/contexta(config)# aaa authentication match TELNET_AUTH outside AuthInbound
```

Configuring Authorization for Network Access

After a user authenticates for a given connection, the FWSM checks for an authorization rule or a dynamic ACL for the traffic. The authorization server or dynamic ACL then determines whether the traffic is allowed or denied.

The FWSM supports TACACS+ authorization servers. You identify the traffic that you want to authorize in the FWSM configuration, and the TACACS+ server determines a user's authorization based on the user profile.

Alternatively, you can use dynamic ACLs that are downloaded from a RADIUS server at the time of authentication. The configuration on the FWSM consists only of the authentication configuration; you enable downloadable ACLs on the server itself.

This section includes the following topics:

- Configuring TACACS+ Authorization, page 12-22
- Configuring RADIUS Authorization, page 12-23

Configuring TACACS+ Authorization

The FWSM lets you configure network access authorization using TACACS+. A user first authenticates using HTTP, Telnet, or FTP. If any traffic from an authenticated user matches an authorization rule, the FWSM sends the username to the TACACS+ server. The TACACS+ server responds to the FWSM with a permit or a deny for that traffic, based on the user's profile. If a user is not yet authenticated, and the traffic matches an authorization statement, then the traffic is blocked. Any traffic that you want to be authorized must also be allowed through the FWSM by an ACL assigned to the interface; you cannot permit addresses in the authorization statement that are denied by the interface ACL.

See the TACACS+ server documentation for information about configuring network access restrictions for a user.

The authorization traffic does not need to be a subset of authentication traffic. Because a user needs to authenticate before authorization occurs, typically the authentication rule includes the same source addresses as the authorization rule. But you can configure a wider authentication rule than authorization, or a wider authorization rule than authentication (for example, a wider range of destination addresses).

To configure authorization, enter the following command:

```
FWSM/contexta(config)# aaa authorization match acl_name interface_name server_group
```

Identify the source addresses and destination addresses using an extended ACL. Create the ACL using the **access-list** command (see the "Adding an Extended Access Control List" section on page 10-13). The **permit** access control entries (ACEs) mark matching traffic for authorization, while **deny** entries exclude matching traffic from authorization.

Note You can alternatively use the **aaa authorization include** command (which identifies traffic within the command). However, you cannot use both methods in the same configuration. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information.

The following commands authenticate and authorize inside Telnet traffic. Telnet traffic to servers other than 209.165.201.5 can be authenticated alone, but traffic to 209.165.201.5 requires authorization:

```
FWSM/contexta(config)# access-list TELNET_AUTH extended permit tcp any any eq telnet
FWSM/contexta(config)# access-list SERVER_AUTH extended permit tcp any host 209.165.201.5
eq telnet
FWSM/contexta(config)# aaa-server AuthOutbound protocol tacacs+
FWSM/contexta(config)# aaa-server AuthOutbound (inside) host 10.1.1.1 TheUauthKey
FWSM/contexta(config)# aaa authentication match TELNET_AUTH inside AuthOutbound
FWSM/contexta(config)# aaa authorization match SERVER_AUTH inside AuthOutbound
```

Configuring RADIUS Authorization

You can configure a RADIUS server to download an ACL or an ACL name to the FWSM at the time of authentication. See the "Configuring Authentication for Network Access" section on page 12-20 for more information about configuring authentication. The user is authorized to do only what is permitted in the user's ACL.

Any traffic that you want to allow according to a dynamic ACL must also be allowed through the FWSM by an ACL assigned to the interface; you cannot permit addresses in the dynamic ACL that are denied by the interface ACL. Because the dynamic ACL is only in place after you authenticate, you should require authentication for all traffic so that the dynamic ACL is in place before any traffic is allowed through the FWSM.

This section includes the following topics:

Configuring the RADIUS Server to Download Per-User Access Control Lists

This section describes how to configure a CiscoSecure ACS RADIUS server or a third-party RADIUS server, and includes the following topics:

Configuring a CiscoSecure ACS RADIUS Server for Downloadable ACLs

You can configure ACLs on the CiscoSecure ACS RADIUS server as a shared profile component and then assign the ACL to a group or to an individual user.

The ACL definition consists of one or more FWSM commands that are similar to the extended **access-list** command (see the "Adding an Extended Access Control List" section on page 10-13), except without the following prefix:

access-list *acl_name* **extended**

The following example is an ACL definition before it is downloaded to the FWSM:

```
+-------------------------------------------+
| Shared profile Components                 |
|                                           |
|   Downloadable PIX ACLs                   |
|                                           |
| Name:               acs_ten_acl           |
| Description:  10 access-list commands  |
|                                           |
|                                           |
|      ACL Definitions                      |
|                                           |
| permit tcp any host 10.0.0.254            |
| permit udp any host 10.0.0.254            |
| permit icmp any host 10.0.0.254           |
| permit tcp any host 10.0.0.253            |
| permit udp any host 10.0.0.253            |
| permit icmp any host 10.0.0.253           |
| permit tcp any host 10.0.0.252            |
| permit udp any host 10.0.0.252            |
| permit icmp any host 10.0.0.252           |
| permit ip any any                         |
+-------------------------------------------+
```

The downloaded ACL on the FWSM has the following name:

#ACSACL#-ip-*acl_name-number*

The *acl_name* argument is the name that is defined on the RADIUS server, and *number* is a unique version ID.

The downloaded ACL on the FWSM consists of the following lines:

```
access-list #ACSACL#-ip-fwsm-acs_ten_acl-3b5385f7 permit tcp any host 10.0.0.254
access-list #ACSACL#-ip-fwsm-acs_ten_acl-3b5385f7 permit udp any host 10.0.0.254
access-list #ACSACL#-ip-fwsm-acs_ten_acl-3b5385f7 permit icmp any host 10.0.0.254
access-list #ACSACL#-ip-fwsm-acs_ten_acl-3b5385f7 permit tcp any host 10.0.0.253
access-list #ACSACL#-ip-fwsm-acs_ten_acl-3b5385f7 permit udp any host 10.0.0.253
access-list #ACSACL#-ip-fwsm-acs_ten_acl-3b5385f7 permit icmp any host 10.0.0.253
access-list #ACSACL#-ip-fwsm-acs_ten_acl-3b5385f7 permit tcp any host 10.0.0.252
access-list #ACSACL#-ip-fwsm-acs_ten_acl-3b5385f7 permit udp any host 10.0.0.252
access-list #ACSACL#-ip-fwsm-acs_ten_acl-3b5385f7 permit icmp any host 10.0.0.252
access-list #ACSACL#-ip-fwsm-acs_ten_acl-3b5385f7 permit ip any any
```

Configuring a Third-Party RADIUS Server for Downloadable ACLs

Configure the ACL using Cisco Vendor Specific Attribute (VSA) number 1 (cisco-AV-pair).

Configure one or more access control entries (ACEs) that are similar to the extended **access-list** command (see the "Adding an Extended Access Control List" section on page 10-13), except that you replace the following command prefix:

access-list *acl_name* **extended**

with the following text:

ip:inacl#*nnn***=**

The *nnn* argument is a number in the range from 0 to 999999999 that identifies the order of the command statement to be configured on the FWSM. If this parameter is omitted, the sequence value is 0, and the order in the RADIUS configuration is used.

The following example is an ACL definition before it is downloaded to the FWSM:

```
ip:inacl#1=permit tcp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
ip:inacl#99=deny tcp any any
ip:inacl#2=permit udp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
ip:inacl#100=deny udp any any
ip:inacl#3=permit icmp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
```

The downloaded ACL name has the following format:

AAA-user-*username*

The *username* argument is the name of the user that is being authenticated.

The downloaded ACL on the FWSM consists of the following lines. Notice the order based on the numbers identified on the RADIUS server

```
access-list  AAA-user-john permit tcp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
access-list  AAA-user-john permit udp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
access-list  AAA-user-john permit icmp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
access-list  AAA-user-john deny tcp any any
access-list  AAA-user-john deny udp any any
```

Downloaded ACLs have two spaces between the word “access-list” and the name. These spaces serve to differentiate a downloaded ACL from a local ACL.

Configuring the RADIUS Server to Download Per-User Access Control List Names

To download a name for an ACL that you already created on the FWSM from the RADIUS server when a user authenticates, configure RADIUS attribute 11 (filter-id) as follows:

filter-id=*acl_name*

See the “Adding an Extended Access Control List” section on page 10-13 to create an ACL on the FWSM.

Configuring Accounting for Network Access

The FWSM can send accounting information to a RADIUS or TACACS+ server about any TCP or UDP traffic that passes through the FWSM. If that traffic is also authenticated, then the AAA server can maintain accounting information by username. If the traffic is not authenticated, the AAA server can maintain accounting information by IP address. Accounting information includes when sessions start and stop, the AAA client messages and username, the number of bytes that pass through the FWSM for the session, the service used, and the duration of each session.

To configure accounting, enter the following command:

```
FWSM/contexta(config)# aaa accounting match acl_name interface_name server_group
```

Identify the source addresses and destination addresses using an extended ACL. Create the ACL using the **access-list** command (see the "Adding an Extended Access Control List" section on page 10-13). The **permit** access control entries (ACEs) mark matching traffic for accounting, while **deny** entries exclude matching traffic from accounting.

Note You can alternatively use the **aaa accounting include** command (which identifies traffic within the command). However, you cannot use both methods in the same configuration. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information.

The following commands authenticate, authorize, and account for inside Telnet traffic. Telnet traffic to servers other than 209.165.201.5 can be authenticated alone, but traffic to 209.165.201.5 requires authorization and accounting:

```
FWSM/contexta(config)# access-list TELNET_AUTH extended permit tcp any any eq telnet
FWSM/contexta(config)# access-list SERVER_AUTH extended permit tcp any host 209.165.201.5
eq telnet
FWSM/contexta(config)# aaa-server AuthOutbound protocol tacacs+
FWSM/contexta(config)# aaa-server AuthOutbound (inside) host 10.1.1.1 TheUauthKey
FWSM/contexta(config)# aaa authentication match TELNET_AUTH inside AuthOutbound
FWSM/contexta(config)# aaa authorization match SERVER_AUTH inside AuthOutbound
FWSM/contexta(config)# aaa accounting match SERVER_AUTH inside AuthOutbound
```

CHAPTER 13

Configuring Application Protocol Inspection

This chapter describes how to use and configure application protocol inspection, which is often called a "fixup." Inspection engines are required for services that embed IP addressing information in the user data packet or that open secondary channels on dynamically assigned ports. These protocols require the Firewall Services Module (FWSM) to do a deep packet inspection instead of passing the packet through the fast path (see the "Stateful Inspection Feature" section on page 1-5 for more information about the fast path). As a result, inspection engines can affect overall throughput.

Several common inspection engines are enabled on the FWSM by default, but you might need to enable others depending on your network. This chapter includes the following sections:

Inspection Engine Overview

This section includes the following topics:

When to Use Application Protocol Inspection

When a user establishes a connection, the FWSM checks the packet against access control lists (ACLs), creates an address translation, and creates an entry for the session in the fast path, so that further packets can bypass time-consuming checks. However, the fast path relies on predictable port numbers and does not perform address translations inside a packet.

Many protocols open secondary TCP or UDP ports. The initial session on a well-known port is used to negotiate dynamically assigned port numbers.

Other applications embed an IP address in the packet that needs to match the source address that is normally translated when it goes through the FWSM.

If you use applications like these, then you need to enable application inspection.

When you enable application inspection for a service that embeds IP addresses, the FWSM translates embedded addresses and updates any checksum or other fields that are affected by the translation.

When you enable application inspection for a service that uses dynamically assigned ports, the FWSM monitors sessions to identify the dynamic port assignments, and permits data exchange on these ports for the duration of the specific session.

Inspection Limitations

See the following limitations for application protocol inspection:

- You can configure up to 32 inspection engines per context. This limit includes the following inspection engines that are enabled by default, making the total number of configurable inspection engines 27: TFTP, Sun RPC over UDP, NetBIOS NameServer, XDMCP, and CUSeeMe. The OraServ and RealAudio inspection engines, which are also enabled by default, do not affect this limit.
- State information for multimedia sessions that require inspection are not passed over the state link for stateful failover.
- For fragmented IP packets, only the first fragment is inspected.
- For segmented TCP packets, if messages are divided between segments, the FWSM cannot inspect the packets.
- Some inspection engines do not support PAT, NAT, policy NAT, outside NAT, or NAT between same security interfaces. See "Inspection Support" for more information about NAT support.

Inspection Support

Table 13-1 describes the inspection engines supported by the FWSM and whether they are compatible with Network Address Translation (NAT), Port Address Translation (PAT), outside NAT, or NAT between same security interfaces. If a inspection engine does not support outside NAT, consider using the **alias** command instead of outside NAT. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information about the **alias** command.

Inspection engines that are enabled for the default port by default are in bold.

Table 13-1 Inspection Engine Support

Application[1]	Configurable	Default Port	NAT Limitations	Comments	Standards[2]
CUSeeMe	No	UDP/7648	No NAT or PAT. Use NAT identity or NAT exemption only.	—	—
DNS over UDP	Yes	UDP/53	No NAT support is available for name resolution through WINS.	No PTR records are changed.	RFC 1123
FTP	Yes	TCP/21	—	—	RFC 1123
H.323 H.225 and **RAS**	Yes	TCP/1720 UDP/1718-1719	No outside NAT. Use the **alias** command. No NAT on same security interfaces.	Does not support segmented messages.	ITU-T H.323, H.245, H225.0, Q.931, Q.932
HTTP	Yes	TCP/80	—	—	RFC 2616
ICMP	Yes	—	—	—	—
ICMP error	Yes	—	—	—	—

Table 13-1 Inspection Engine Support (continued)

Application[1]	Configurable	Default Port	NAT Limitations	Comments	Standards[2]
ILS (LDAP)	Yes	TCP/389	No outside NAT. Use the **alias** command. No PAT.	—	—
MGCP	Yes	UDP/2427, 2727	No NAT or PAT. Use NAT identity or NAT exemption only.	—	RFC2705bis-05
NetBIOS Name Server over IP	No	UDP/137-138	—	—	—
OraServ	No	UDP/1525	—	—	
RealAudio	No	UDP/7070	—	—	
RSH	Yes	TCP/514	No PAT.	—	Berkeley UNIX
RTSP	Yes	TCP/554	No PAT. No outside NAT. Use the **alias** command.	No handling for HTTP cloaking.	RFC 2326, RFC 2327, RFC 1889
SIP TCP	Yes	TCP/5060	No outside NAT. Use the **alias** command. No NAT on same security interfaces.	—	RFC 2543
SIP UDP	Yes	UDP/5060	No outside NAT. Use the **alias** command. No NAT on same security interfaces.	—	RFC 2543
SKINNY (SCCP)	Yes	TCP/2000	No outside NAT. Use the **alias** command. No NAT on same security interfaces.	Does not handle TFTP uploaded Cisco IP Phone configurations.	—
SMTP	Yes	TCP/25	—	—	RFC 821, 1123
SQL*Net	Yes	TCP/1521 (v1)	No policy NAT.	v1 and v2.	—
Sun RPC over UDP	No	UDP/111	No NAT or PAT. Use NAT identity or NAT exemption only.	—	—
Sun RPC over TCP	Yes	TCP/111	No NAT or PAT. Use NAT identity or NAT exemption only.	—	—
TFTP	No	UDP/69	Payload IP address not translated.	—	RFC 1350
XDMCP	No	UDP/177	No NAT or PAT. Use NAT identity or NAT exemption only.	—	—

1. Inspection engines that are enabled by default for the default port are in bold.
2. The FWSM is in compliance with these standards, but it does not enforce compliance on packets being inspected. For example, FTP commands are supposed to be in a particular order, but the FWSM does not enforce the order.

Configuring an Inspection Engine

Disabling or modifying an inspection engine only affects connections that are initiated after the command is processed. Disabling an inspection engine for a specific port or application does not affect existing connections. If you want the change to take effect immediately, enter the **clear xlate** command to remove all existing sessions.

To configure an inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol {
dns [maximum-length length]
ftp [strict] [port[-port]] |
h323 {h225 | ras} [port[-port]] |
http [port[-port]] |
icmp |
icmp error |
ils [port[-port]] |
mgcp [port[-port] |
rpc [port[-port] |
rsh [port[-port] |
rtsp [port[-port] |
sip [port[-port] |
sip udp |
skinny [port[-port] |
smtp [port[-port]] |
sqlnet [port[-port]]}
```

For most applications and protocols, you can define multiple port assignments, which is useful when multiple instances of the same service are running on different ports.

Because you can enter multiple ports (either as a range or as separate commands), if you specify a new port, that port is added to the configuration along with previously configured ports. To remove a port, enter the **no** version of the command.

See the following keywords:

- **dns maximum-length** *length*—This option sets the maximum length of a DNS reply. The default is 512 bytes. This inspection engine uses UDP port 53, and the port is not configurable.
- **ftp strict**—This option only lets an FTP server generate the 227 command and only lets an FTP client generate the PORT command. The 227 and PORT commands are checked to ensure they do not appear in an error string. This limitation prevents clients from sending embedded commands in FTP requests. Each FTP command must be acknowledged before a new command is allowed.
- **h323** {**h225** | **ras**}—You can set the inspection engines for H.323 and RAS (**h225** and **ras**) separately.

See the "Detailed Information About Inspection Engines" section on page 13-5 for information about each protocol inspection engine.

By default, an inspection engine for FTP port 21 is enabled. The following example shows how to define additional ports for FTP:

```
FWSM/contexta(config)# fixup protocol ftp 2100
FWSM/contexta(config)# fixup protocol ftp 4254
FWSM/contexta(config)# fixup protocol ftp 9090
```

After entering these commands, the FWSM listens for FTP traffic on port 21, as well as 2100, 4254, and 9090.

The following command assigns the port range from 1500 to 2000 to SQL*Net:

```
FWSM/contexta(config)# fixup protocol sqlnet 1500-2000
```

Detailed Information About Inspection Engines

CUSeeMe Inspection Engine

Enabled by default for UDP port 7648

Not Configurable

With CUSeeMe clients, one user can connect directly to another (CUSeeMe or other H.323 client) for person-to-person audio, video, and data collaboration. CUSeeMe clients can conference in a mixed client environment that includes both CUSeeMe clients and H.323-compliant clients from other vendors.

Behind the scenes, CUSeeMe clients operate in two different modes. When connected to another CUSeeMe client or CUSeeMe Conference Server, the client sends information in CUSeeMe mode.

When connected to an H.323-compliant videoconferencing client from a different vendor, CUSeeMe clients communicate using the H.323-standard format in H.323 mode.

CUSeeMe is supported through H.323 inspection, as well as performing NAT on the CUSeeMe control stream, which operates on UDP port 7648.

DNS over UDP Inspection Engine

Enabled by default for UDP port 53

Domain Name System (DNS) requests require an inspection engine so that DNS queries are not subject to the generic UDP handling based on activity timeouts. Instead, the UDP connections associated with DNS queries and responses are torn down as soon as a reply to a DNS query has been received. The DNS inspection engine monitors the message exchange to ensure that the ID of the DNS reply matches the ID of the DNS query. See the "DNS and NAT" section on page 9-13 for more information about how the FWSM alters the DNS payload.

This functionality is different from DNS Guard. See the "Other Protection Features" section on page 1-6 for more information about DNS Guard.

To configure the maximum length of the DNS reply, enter the following command:

```
FWSM/contexta(config)# fixup protocol dns [maximum-length length]
```

The default is 512 bytes. The port is 53 (UDP) and is not configurable.

FTP Inspection Engine

Enabled by default for TCP port 21

To configure the FTP inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol ftp [strict] port[-port]
```

The default port is 21 (TCP).

If you disable FTP inspection engines with the **no fixup protocol ftp** command, outbound users can start connections only in passive mode, and inbound users can start connections only in active mode.

The FTP inspection engine inspects the FTP sessions, and performs four tasks:

- Prepares dynamic secondary data connection—The channels are allocated in response to a file upload, a file download, or a directory listing event and must be pre-negotiated. The port is negotiated through the PORT or PASV commands.
- Tracks **ftp** command-response sequence—If the **strict** option is enabled, each **ftp** command and response sequence is tracked for the following anomalous activity:
  - Truncated command—Number of commas in the PORT and PASV reply command is checked to see if it is five. If it is not five, then the PORT command is assumed to be truncated and the TCP connection is closed.
  - Incorrect command—Checks the **ftp** command to see if it ends with <CR><LF> characters, as required by the RFC. If it does not, the connection is closed.
  - Size of RETR and STOR commands—These are checked against a fixed constant of 256. If the size is greater, then an error message is logged and the connection is closed.
  - Command spoofing—The PORT command should always be sent from the client. The TCP connection is denied if a PORT command is sent from the server.

  - Reply spoofing—PASV reply command (227) should always be sent from the server. The TCP connection is denied if a PASV reply command is sent from the client. This prevents the security hole when the user executes "227 xxxxx a1, a2, a3, a4, p1, p2."
  - TCP stream editing.
  - Invalid port negotiation—The negotiated dynamic port value is checked to see if it is less than 1024. As port numbers in the range from 1 to 1024 are reserved for well-known connections, if the negotiated port falls in this range, then the TCP connection is freed.
  - Command pipelining—The number of characters present after the port numbers in the PORT and PASV reply command is cross checked with a constant value of 8. If it is more than 8, then the TCP connection is closed.

Note The use of the **strict** option may break FTP clients that do not comply with the RFC standards.

- Generates an audit trail—The FTP inspection engine generates the following system messages:
  - System message 303002 is generated for each file that is retrieved or uploaded.
  - System message 201005 is generated if the secondary dynamic channel preparation failed due to memory shortage.
- Translates embedded IP addresses—In conjunction with NAT, the FTP inspection engine translates the IP address within the application payload. This is described in detail in RFC 959.

H.323 Inspection Engine

H.323 H.225 enabled by default for TCP port 1720

H.323 RAS enabled by default for UDP ports 1718-1719

The **fixup protocol h323** command provides support for H.323-compliant endpoints. The FWSM supports H.323 Version 1, 2, 3, and 4.

H.323 is a suite of protocols defined by the International Telecommunication Union (ITU) for multimedia conferences over LANs. H.323 supports VoIP gateways and VoIP gatekeepers.

This section includes the following topics:

Configuring the H.323 Inspection Engine

To configure the H.323 inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol h323 {h225 | ras} [port[-port]]
```

You can set the inspection engines for H.232 and RAS (**h225** and **ras**) separately. The default port for **h225** is 1720 (TCP), and the default ports for **ras** are 1718-1719 (UDP).

Multiple Calls on One Call Signalling Connection

Allowing multiple calls on the same call signaling channel reduces call setup time and reduces the use of ports on the FWSM.

To configure how long the H.225 call signaling channel stays open, enter the following command:

```
FWSM/contexta(config)# timeout h225 hh[:mm[:ss]]
```

The default is 1 hour.

For example, to keep the channel open without any timeout, set the timer to 0 by entering the following command:

```
timeout h225 00:00:00
```

To disable the timer and close the TCP connection immediately after all calls are cleared, set the timeout value to 1 second, as follows:

```
timeout h225 00:00:01
```

Viewing Connection Status

To display the status of H.225 connections, enter the following command:

```
FWSM/contexta(config)# show conn state h225
```

Technical Background

The H.323 collection of protocols collectively can use up to two TCP connections and four to six UDP connections. FastConnect uses only one TCP connection, and RAS uses a single UDP connection for registration, admissions, and status.

An H.323 client might initially establish a TCP connection to an H.323 server using TCP port 1720 to request Q.931 call setup. As part of the call setup process, the H.323 terminal supplies a port number to the client to use for an H.245 TCP connection.

Note In environments where an H.323 gatekeeper is in use, the initial packet is transmitted using UDP.

The H.323 inspection engine monitors the Q.931 TCP connection to determine the H.245 port number. If the H.323 terminals are not using FastConnect, the FWSM dynamically allocates the H.245 connection based on the inspection of the H.225 messages.

Within each H.245 message, the H.323 endpoints exchange port numbers that are used for subsequent UDP data streams. The H.323 inspection engine inspects the H.245 messages to identify these ports and dynamically creates connections for the media exchange. Real-Time Transport Protocol (RTP) uses the negotiated port number, while RTP Control Protocol (RTCP) uses the next higher port number.

The H.323 control channel handles H.225 and H.245 and H.323 RAS. The H.323 inspection engine uses the following ports:

- 1718—Gate Keeper Discovery UDP port
- 1719—RAS UDP port
- 1720—TCP Control Port

The two major functions of the H.323 inspection engine are as follows:

- NAT the necessary embedded IPv4 addresses in the H.225 and H.245 messages. Because H.323 messages are encoded in PER encoding format, FWSM uses an ASN.1 decoder to decode the H.323 messages. The H.323 inspection engine supports static NAT and dynamic NAT. It does not support NAT on same security interfaces or outside NAT.
- Dynamically allocate the negotiated H.245 and RTP/RTCP connections.

The FWSM administrator must configure an access control list (ACL) for the well-known H.323 port 1720 for the H.225 call signaling. However, the H.245 signaling ports are negotiated between the endpoints in the H.225 signaling.

Note When an H.323 gatekeeper is used, the FWSM opens an H.225 connection based on inspection of the AdmissionConfirm (ACF) message.

The FWSM dynamically allocates the H.245 channel after inspecting the H.225 messages and then "hooks up" the H.245 channel to be fixed up as well. That means whatever H.245 messages pass through the FWSM are passed through the H.245 inspection engine, NATing embedded IP addresses and opening the negotiated media channels.

The H.323 ITU standard requires that a TPKT header, defining the length of the message, precede the H.225 and H.245, before being passed on to the reliable connection. Because the TPKT header does not necessarily need to be sent in the same TCP packet as the H.225/H.245 message, the FWSM must remember the TPKT length to process/decode the messages properly. FWSM keeps a data structure for each connection and that data structure contains the TPKT length for the next expected message.

If the FWSM needs to NAT any IP addresses, then it will have to change the checksum, the UUIE (user-user information element) length, and the TPKT, if included in the TCP packet with the H.225 message. If the TPKT is sent in a separate TCP packet, then the FWSM will proxy ACK that TPKT and append a new TPKT to the H.245 message with the new length.

Note The FWSM does not support TCP options in the Proxy ACK for the TPKT.

Each UDP connection with a packet going through the H.323 inspection engine is marked as an H.323 connection and will time out with the H.323 timeout as configured by the administrator using the **timeout** command.

HTTP Inspection Engine

The HTTP inspection engine enables the system message 304001 when an inside user issues an HTTP GET request:

%FWSM-5-304001: *user source_address* Accessed [JAVA] URL *dest_address*: *url*.

To configure the HTTP inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol http [port[-port]]
```

The default port is 80 (TCP).

ICMP Inspection Engine

The ICMP inspection engine allows ICMP traffic to have a "session" so it can be inspected like TCP and UDP traffic. Without the ICMP inspection engine, we recommend that you do not allow ICMP through the FWSM in an ACL. Without stateful inspection, ICMP can be used to attack your network. The ICMP inspection engine ensures that there is only one response for each request, and that the sequence number is correct.

To configure the ICMP inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol icmp
```

The ICMP payload is scanned to retrieve the five-tuple from the original packet. The ICMP inspection engine supports both one-to-one NAT and PAT. Using the retrieved five-tuple, a lookup is performed to determine the original address of the client. The ICMP inspection engine makes the following changes to the ICMP packet:

- In the IP Header, the NAT IP is changed to the Client IP (Destination Address) and the IP checksum is modified.
- In the ICMP Header, the ICMP checksum is modified due to the changes in the ICMP packet.
- In the Payload, the following changes are made:
  - Original packet NAT IP is changed to the Client IP
  - Original packet NAT port is changed to the Client Port
  - Original packet IP checksum is recalculated

ICMP Error Inspection Engine

The FWSM supports NAT of ICMP error messages. When this feature is enabled, the FWSM creates translation sessions for intermediate hops that send ICMP error messages, based on the NAT configuration. The FWSM overwrites the packet with the translated IP addresses.

To configure the ICMP error inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol icmp error
```

When disabled, the FWSM does not create translation sessions for intermediate nodes that generate ICMP error messages. ICMP error messages generated by the intermediate nodes between the inside host and the FWSM reach the outside host without consuming any additional NAT resource. This is undesirable when an outside host uses the **traceroute** command to trace the hops to the destination on the inside of the FWSM. When the FWSM does not NAT the intermediate hops, all the intermediate hops appear with the translated destination IP address.

ILS Inspection Engine

Enabled by default for TCP port 389

The Internet Locator Service (ILS) is based on the Lightweight Directory Access Protocol (LDAP) and is LDAPv2 compliant. ILS was developed by Microsoft for use with its NetMeeting, SiteServer, and Active Directory products.

To configure the ILS inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol ils [port[-port]]
```

The default port is 389 (TCP).

The FWSM supports NAT for ILS, which is used to register and locate endpoints in the ILS or SiteServer Directory. PAT is not supported because only IP addresses, and not ports, are stored by an LDAP database.

For search responses, when the LDAP server is located outside, NAT should be considered to allow internal peers to communicate locally while registered to external LDAP servers. For such search responses, translation sessions are searched first, and then NAT entries to obtain the correct address. If both of these searches fail, then the address is not changed.

Note For sites using NAT exemption or identity NAT, we recommend that you disable this inspection engine for better performance.

Additional configuration might be necessary when the ILS server is located inside the FWSM border. This requires an ACL for outside clients to access the LDAP server on the specified port, typically TCP 389.

ILS/LDAP follows a client/server model with sessions handled over a single TCP connection. Depending on the client's actions, several of these sessions might be created.

During connection negotiation time, a Berkeley Internet Name Domain (BIND) protocol data unit (PDU) is sent from the client to the server. Once a successful BIND RESPONSE from the server is received, other operational messages might be exchanged (such as ADD, DEL, SEARCH, or MODIFY) to perform operations on the ILS Directory. The ADD REQUEST and SEARCH RESPONSE PDUs might contain IP addresses of NetMeeting peers, used by H.323 (SETUP and CONNECT messages) to establish NetMeeting sessions. Microsoft NetMeeting v2.X and v3.X provide ILS support.

The ILS inspection engine performs the following operations:

- Decodes the LDAP REQUEST/RESPONSE PDUs using the bit error rate (BER) decode functions
- Parses the LDAP packet
- Extracts IP addresses
- Translates IP addresses as necessary
- Encodes the PDU with translated addresses using BER encode functions
- Copies the newly encoded PDU back to the TCP packet
- Performs incremental TCP checksum and sequence number adjustment

The ILS inspection engine has the following limitations:

- Referral requests and responses are not supported
- Users in multiple directories are not unified
- Single hosts that register to multiple directories using different name are not supported by the ILS inspection engine. You must use the same for all directories.

MGCP Inspection Engine

The Media Gateway Control Protocol (MGCP) is used for controlling media gateways from external call control elements called media gateway controllers, or call agents. A media gateway is typically a network element that provides conversion between the audio signals carried on telephone circuits and data packets carried over the Internet or over other packet networks.

To use MGCP, you typically need to configure at least two ports. One on which the gateway receives commands and one for the port on which the call agent receives commands. Normally, a call agent sends commands to port 2427, while a gateway sends commands to port 2727.

To configure the MGCP inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol mgcp port[-port]
```

The default ports are 2427 and 2727

Neither NAT or PAT are supported by the FWSM with MGCP.

This section includes the following topics:

MGCP Overview

Examples of media gateways are:

- Trunking gateways, that interface between the telephone network and a Voice over IP network. Such gateways typically manage a large number of digital circuits.
- Residential gateways, that provide a traditional analog (RJ11) interface to a Voice over IP network. Examples of residential gateways include cable modem/cable set-top boxes, xDSL devices, broad-band wireless devices.
- Business gateways, that provide a traditional digital PBX interface or an integrated soft PBX interface to a Voice over IP network.

MGCP messages are transmitted over UDP. A response is sent back to the source address (IP address and UDP port number) of the command, but the response may not arrive from the same address as the command was sent to. This can happen when multiple call agents are being used in a failover configuration and the call agent that received the command has passed control to a backup call agent, which then sends the response.

Configuration for Multiple Call Agents and Gateways

Use the following commands to configure the FWSM to support the use of multiple MGCP call agents and gateways:

- To specify a group of call agents that can manage one or more gateways, enter the following command:

  ```
  FWSM/contexta(config)# mgcp call-agent ip_address group_id
  ```

  This information is used to open connections for the other call agents than the one a gateway sends a command to, so that any of the call agents can send the response. The *ip_address* specifies the IP address of the call agent. The *group_id* is a number from 0 to 4294967295. Call agents with the same *group_id* belong to the same group.

- To specify the maximum number of MGCP commands that can be queued waiting for a response, enter the following command:

  ```
  FWSM/contexta(config)# mgcp command-queue limit
  ```

  The range of allowed values for *limit* is 1 to 4294967295. The default is 200. When the limit is reached and a new command arrives, the command that was in the queue for the longest time is removed.

- To specify which group of call agents are managing a particular gateway, enter the following command:

  ```
  FWSM/contexta(config)# mgcp gateway ip_address group_id
  ```

  The IP address of the gateway is specified with the *ip_address* option. The *group_id* option is a number from 0 to 4294967295. It must correspond with the *group_id* of the call agents that are managing the gateway.

The following example limits the MGCP command queue to 150 commands, allows call agents 10.10.11.5 and 10.10.11.6 to control gateway 10.10.10.115 and allows call agents 10.10.11.7 and 10.10.11.8 to control gateway 10.10.10.116:

```
FWSM/contexta(config)# mgcp call-agent 10.10.11.5 101
FWSM/contexta(config)# mgcp call-agent 10.10.11.6 101
FWSM/contexta(config)# mgcp call-agent 10.10.11.7 102
FWSM/contexta(config)# mgcp call-agent 10.10.11.8 102
```

```
FWSM/contexta(config)# mgcp command-queue 150
FWSM/contexta(config)# mgcp gateway 10.10.10.115 101
FWSM/contexta(config)# mgcp gateway 10.10.10.116 102
```

Viewing MGCP Information

- To view information about MGCP, enter the following command:

  ```
  FWSM/contexta(config)# show mgcp {commands | sessions} [detail]
  ```

  Use the **commands** option to list the commands in the command queue. Use the **sessions** option to list the existing MGCP sessions. Use the **detail** option to list detailed information about each command or session.

- To show information about the MGCP connections, enter the following command:

  ```
  FWSM/contexta(config)# show conn {detail | state} mgcp
  ```

  Use the **detail** option to display detailed information about the MGCP connections. Use the **state** option to display the media connections created for MGCP sessions.

NetBios Name Service Inspection Engine

Enabled by default for UDP ports 137 and 138

Not Configurable

The NetBios inspection engine translates IP addresses in the NetBios name service (NBNS) packets according to the FWSM NAT configuration.

OraServ Inspection Engine

Enabled by default for UDP port 1525

Not Configurable

The OraServ inspection engine allows the data channel to go through the FWSM.

RealAudio Inspection Engine

Enabled by default for UDP port 7070

Not Configurable

The RealAudio inspection engine allows the data channel to go through the FWSM when the data channel source port is between UDP ports 6790 and 7170.

RSH Inspection Engine

Enabled by default for TCP port 514

The Remote Shell (RSH) protocol uses a TCP connection from the RSH client to the RSH server on TCP port 514. The client and server negotiate the TCP port number where the client will listen for the STDERR output stream. The RSH inspection engine supports NAT of the negotiated port number if necessary.

To configure the RSH inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol rsh [port[-port]]
```

The default port for the initial RSH connection is 514 (TCP).

RTSP Inspection Engine

Real Time Streaming Protocol (RTSP) is used by RealAudio, RealNetworks, Apple QuickTime 4, RealPlayer, and Cisco IP/TV connections. FWSM does not support multicast RTSP.

RTSP applications use the well-known port 554 with TCP (rarely UDP) as a control channel. The FWSM only supports TCP, in conformity with RFC 2326.

This TCP control channel is used to negotiate the data channels that are used to transmit audio/video traffic, depending on the transport mode that is configured on the client.

The supported Real Data Transports (RDTs) are: rtp/avp, rtp/avp/udp, x-real-rdt, x-real-rdt/udp, and x-pn-tng/udp.

The FWSM parses Setup response messages with a status code of 200. If the response message is travelling inbound, the server is outside relative to the FWSM and dynamic channels need to be opened for connections coming inbound from the server. If the response message is outbound, then the FWSM does not need to open dynamic channels.

Because RFC 2326 does not require that the client and server ports must be in the SETUP response message, the FWSM will need to keep state and remember the client ports in the SETUP message. QuickTime places the client ports in the SETUP message and then the server responds with only the server ports.

To configure the RTSP inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol rtsp [port[-port]]
```

The default port is 554 (TCP).

If you are using Cisco IP/TV, use RTSP TCP port 554 and TCP 8554 as follows:

```
FWSM/contexta(config)# fixup protocol rtsp 554
FWSM/contexta(config)# fixup protocol rtsp 8554
```

The following restrictions apply to the RTSP inspection engine:

- The FWSM does not inspect RTSP messages passing through UDP ports.
- The FWSM does not inspect inbound RTSP connections.

- The FWSM does not support RealNetworks multicast mode (x-real-rdt/mcast).
- The FWSM does not support PAT and outside NAT for RTSP.
- The FWSM does not have the ability to recognize HTTP cloaking where RTSP messages are hidden in HTTP messages.
- The FWSM cannot perform NAT on RTSP messages because the embedded IP addresses are contained in the Session Description Protocol (SDP) files as part of HTTP or RTSP messages. Packets could be fragmented, and the FWSM cannot perform NAT on fragmented packets.
- With Cisco IP/TV, the number of translations the FWSM performs on the SDP part of the message is proportional to the number of program listings in the Content Manager (each program listing can have at least six embedded IP addresses).
- You can configure NAT for Apple QuickTime 4 or RealPlayer. Cisco IP/TV only works with NAT if the Viewer and Content Manager are on the outside network and the server is on the inside network.
- When using RealPlayer, it is important to properly configure transport mode. For the FWSM, add an **access-list** command statement from the server to the client or vice versa. For RealPlayer, change transport mode by clicking **Options>Preferences>Transport>RTSP Settings**.

  If you use TCP mode on the RealPlayer, select the **Use TCP to Connect to Server** and **Attempt to use TCP for all content** check boxes. On the FWSM, there is no need to configure the inspection engine.

  If you use UDP mode on the RealPlayer, select the **Use TCP to Connect to Server** and **Attempt to use UDP for static content** check boxes. On the FWSM, configure the RTSP inspection engine.

SIP Inspection Engine

Enabled by default for TCP and UDP port 5060

Session Initiation Protocol (SIP), as defined by the Internet Engineering Task Force (IETF), enables call handling sessions, particularly two-party audio conferences, or “calls.”

This section includes the following topics:

Configuring the SIP Inspection Engine

To configure the SIP inspection engine, enter the following commands:

- To configure the SIP TCP inspection engine, enter the following command:

  ```
  FWSM/contexta(config)# fixup protocol sip [port[-port]]
  ```

  The default port is 5060 (TCP).

- To configure the SIP UDP inspection engine, enter the following command:

  ```
  FWSM/contexta(config)# fixup protocol sip udp
  ```

  The default port is 5060 (UDP), which is the only port allowed.

SIP Overview

SIP works with Session Description Protocol (SDP) for call signalling. SDP specifies the ports for the media stream. The inspection engine supports the following SIP message types. Other message types are allowed through the FWSM, but they are not inspected.

- Messages in RFC 2543 (redefined in RFC 3261):
  - INVITE
  - ACK
  - BYE
  - CANCEL
  - REGISTER
  - Responses 1xx, 2xx, 3xx, 4xx, 5xx, 6xx
- Message in RFC 2976:
  - INFO
- Messages in RFC 3265:
  - SUBSRIBE
  - NOTIFY
- Message in RFC 3428:
  - MESSAGE

To support SIP calls through the FWSM, the FWSM inspects signaling messages for the media connection addresses, media ports, and embryonic connections for the media, because while the signaling is sent over a well-known destination port (UDP/TCP 5060), the media streams are dynamically allocated. Also, SIP embeds IP addresses in the user-data portion of the IP packet. The SIP inspection engine applies NAT for these embedded IP addresses. It does not support NAT between same security interfaces or outside NAT.

Technical Background

The SIP inspection engine NATs the SIP text-based messages, recalculates the content length for the SDP portion of the message, and recalculates the packet length and checksum. It dynamically opens media connections for ports specified in the SDP portion of the SIP message as address/ports on which the endpoint should listen.

The SIP inspection engine has a database that keeps track of information from the SIP payload that identifies the call, as well as the source and destination. Contained within this database are the media addresses and media ports that were contained in the SDP media information fields and the media type. There can be multiple media addresses and ports for a session. RTP/RTCP connections are opened between the two endpoints using these media addresses/ports. The well-known port 5060 must be used on the initial call setup (INVITE) message. However, subsequent messages may not have this port number. The SIP inspection engine opens signaling connection pinholes, and marks these connections as SIP connections. This is done for the messages to reach the SIP application and be NATed.

As a call is set up, the SIP session is considered in the “transient” state until the media address and media port is received in a Response message from the called endpoint indicating the RTP port the called endpoint will listen on. If there is a failure to receive the response messages within one minute, the signaling connection will be torn down.

Once the final handshake is made, the call state is moved to active and the signaling connection will remain until a BYE message is received.

If an inside endpoint initiates a call to an outside endpoint, a media hole is opened to the outside interface to allow RTP/RTCP UDP packets to flow to the inside endpoint media address and media port specified in the INVITE message from the inside endpoint. Unsolicited RTP/RTCP UDP packets to an inside interface will not traverse the FWSM, unless the FWSM configuration specifically allows it.

The media connections are torn down within two minutes after the connection becomes idle. This is, however, a configurable timeout and can be set for a shorter or longer period of time. See the **timeout** command in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

Skinny Inspection Engine

Enabled by default for TCP port 2000

Skinny (or Simple) Client Control Protocol (SCCP) is a protocol used in VoIP networks.

To configure the Skinny inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol skinny [port[-port]]
```

The default port is 2000 (TCP).

This section includes the following topics:

Skinny Overview

Cisco IP Phones using Skinny can coexist with an H.323 environment. When used with Cisco CallManager, the Skinny client can interoperate with H.323-compliant terminals. The FWSM ensures that all SCCP signalling and media packets can traverse the FWSM by providing NAT of the SCCP Signaling packets. This inspection engine does not support NAT between same security interfaces.

There are 5 versions of the SCCP protocol supported: 2.4, 3.0.4, 3.1.1, 3.2, and 3.3.2.

The FWSM supports DHCP options 150 and 66, which allow the FWSM to send the location of a TFTP server to Cisco IP Phones and other DHCP clients. The TFTP server provides the address of the Cisco CallManager for the Cisco IP Phones. For further information about this feature, see the "Configuring the DHCP Server" section on page 8-19. If the Cisco CallManager is on a higher security interface, which requires NAT for the Cisco CallManager IP address, and you configure the TFTP server to serve a file with the local untranslated address of the Cisco CallManager, then the Cisco IP Phones cannot contact the Cisco CallManager. We recommend that you use the Cisco CallManager name instead of the IP address, and rely on the DNS server to provide the correct address. If the DNS server is also on the higher security interface, the FWSM can use the DNS inspection engine to translate the address inside the DNS response.

If you enter the **clear xlate** command after PAT translations are created for Cisco CallManager, Skinny calls cannot be established because the translations for the Cisco CallManager are permanently deleted. Under these circumstances, Cisco IP Phones need to reregister with the Cisco CallManager to establish calls through the FWSM.

Problems with Fragmented Skinny Packets

The FWSM does not correctly handle fragmented Skinny packets. For instance, when using a voice conferencing bridge, Skinny packets might become fragmented and are then dropped by the FWSM. This happens because the Skinny inspection engine checks each packet and drops what appear to be bad packets. When a single Skinny packet is fragmented into multiple TCP packets, the Skinny inspection engine finds that the internal checksums within the Skinny packet fragments are not correct and so it drops the packet.

SMTP Inspection Engine

Enabled by default for TCP port 25

The SMTP inspection engine enables the Mail Guard feature. This restricts mail servers to receiving the seven minimal commands defined in RFC 821, section 4.5.1 (HELO, MAIL, RCPT, DATA, RSET, NOOP, and QUIT). All other commands are rejected.

Microsoft Exchange server does not strictly comply with RFC 821 section 4.5.1, using extended SMTP commands such as EHLO. The FWSM converts any such commands into NOOP commands, which as specified by the RFC, forces SMTP servers to fall back to using minimal SMTP commands only. This might cause Microsoft Outlook clients and Exchange servers to function unpredictably when their connection passes through FWSM. In this case, you might want to disable the SMTP inspection engine, although the Mail Guard feature does provide valuable protection.

To configure the SMTP inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol smtp [port[-port]]
```

The default port is 25 (TCP).

An SMTP server responds to client requests with numeric reply codes and optional human-readable strings. The SMTP inspection engine controls and reduces the commands that the user can use as well as the messages that the server returns. The SMTP inspection engine performs three primary tasks:

- Restricts SMTP requests to seven minimal commands (HELO, MAIL, RCPT, DATA, RSET, NOOP, and QUIT).
- Changes the characters in the server SMTP banner to asterisks except for the “2”, “0”, “0” characters. Carriage return (CR) and linefeed (LF) characters are ignored.
- Monitors the SMTP command-response sequence.
- Generates an audit trail—Audit record 108002 is generated when an invalid character embedded in the mail address is replaced. For more information, see RFC 821.

The SMTP inspection engine monitors the command and response sequence for the following anomalous signatures:

- Truncated commands.
- Incorrect command termination (not terminated with <CR><LR>).
- The MAIL and RCPT commands specify who are the sender and the receiver of the mail. Mail addresses are scanned for strange characters. The pipeline character (|) is deleted (changed to a blank space) and “<” ,”>” are only allowed if they are used to define a mail address (“>” must be preceded by “<”).

- Unexpected transition by the SMTP server.
- For unknown commands, the FWSM changes all the characters in the packet to X. In this case, the server will generate an error code to the client. Because of the change in the packet, the TCP checksum has to be recalculated.
- TCP stream editing.
- Command pipelining.

SQL*Net Inspection Engine

Enabled by default for TCP port 1521

The SQL*Net protocol consists of different packet types that the FWSM handles to make the data stream appear consistent with the Oracle applications on either side of the FWSM.

To configure the SQL*Net inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol sqlnet [port[-port]]
```

The default port is 1521 (TCP).

The FWSM NATs all addresses and looks in the packets for all embedded ports to open for SQL*Net Version 1.

For SQL*Net Version 2, all DATA or REDIRECT packets that immediately follow REDIRECT packets with a zero data length are fixed up.

The packets that need inspection engine contain embedded host/port addresses in the following format:

```
(ADDRESS=(PROTOCOL=tcp)(DEV=6)(HOST=a.b.c.d)(PORT=a))
```

SQL*Net Version 2 TNSFrame types (Connect, Accept, Refuse, Resend, and Marker) are not scanned for addresses to NAT, nor does the inspection engine open dynamic connections for any embedded ports in the packet.

SQL*Net Version 2 TNSFrames, Redirect, and Data packets are scanned for ports to open and addresses to NAT, if preceded by a REDIRECT TNSFrame type with a zero data length for the payload. When the Redirect message with data length zero passes through the FWSM, a flag is set in the connection data structure to expect the Data or Redirect message that follows is NATed and ports are dynamically opened. If one of the TNS frames in the preceding paragraph arrives after the Redirect message, the flag is reset.

The SQL*Net inspection engine recalculates the checksum, change IP, TCP lengths, and readjusts Sequence Numbers and Acknowledgment Numbers using the delta of the length of the new and old message.

SQL*Net Version 1 is assumed for all other cases. TNSFrame types (Connect, Accept, Refuse, Resend, Marker, Redirect, and Data) and all packets are scanned for ports and addresses. Addresses are NATed and port connections are opened.

Sun RPC Inspection Engine

Enabled by default for UDP port 111

Sun Remote Procedure Call (RPC) is used by many services, for example, Network File System (NFS) and Network Information Service (NIS).

Sun RPC services can run on any port on the system. When a client attempts to access an RPC service on a server, it must find out which port that service is running on. It does this by querying the portmapper process on the well-known port of 111.

The client sends the RPC program number of the service, and gets back the port number. From this point on, the client program sends its RPC queries to that new port.

When a server sends out a reply, the FWSM intercepts this packet and opens both embryonic TCP and UDP connections on that port for a short period of time. After the client connects to the port and makes a full connection, the embryonic connection goes away. For additional connections from the client to the port, the client must repeat the portmapper process. Alternatively, you can configure the FWSM to keep the embryonic connections open for a longer period of time so that clients can use cached port numbers and do not have to repeat the portmapper process. This method is required for Sun RPC over TCP; only the default inspection for UDP uses the above method. See the **rpc-server** command below.

NAT or PAT of RPC payload information is not supported. Use NAT exemption or identity NAT.

- To configure the Sun RPC inspection engine for TCP, enter the following command:

  ```
  FWSM/contexta(config)# fixup protocol rpc [port[-port]]
  ```

  The default port is 111 (TCP). You must also configure the **rpc-server** command (below). The UDP inspection engine is on by default and is not configurable.

- To allow clients to use cached port numbers for Sun RPC services (such as NFS or NIS), enter the following command:

  ```
  FWSM/contexta(config)# rpc-server interface_name ip_address mask service service_type
  protocol {tcp | udp} port[-port] timeout hh:mm:ss
  ```

  After a client initially connects to a server running a Sun RPC service, the client might cache the Sun RPC port information supplied by the portmapper process. Additional connections from the client might use these cached ports. This command allows clients to use cached port numbers for the duration of the specified **timeout** rather than have to re-request the port numbers from the portmapper process. This command is required for Sun RPC over TCP.

TFTP Inspection Engine

Enabled by default for UDP port 69

Not Configurable

The FWSM permits all UDP connections from a TFTP server back to a client source port if there is an existing TFTP connection between the server and client.

XDMCP Inspection Engine

Enabled by default for UDP port 177

Not Configurable

The port assignment for the X Display Manager Control Protocol (XDMCP) is not configurable. XDMCP is a protocol that uses UDP port 177 to negotiate X sessions, which use TCP when established.

For successful negotiation and as the start of an Xwindows session, the FWSM must allow the TCP back connection. Once XDMCP negotiates the session, a single embryonic connection is created to handle the initial TCP connection, after which the established rule is consulted.

During the X Windows session, the manager talks to the display's Xserver on the well-known port 6000 + n. Each display has a separate connection to the Xserver as a result of the following terminal setting:

```
setenv DISPLAY Xserver:n
```

where *n* is the display number.

When XDMCP is used, the display is negotiated using IP addresses, which the FWSM can NAT if needed. The XDCMP inspection engine does not support PAT.

Filtering HTTP, HTTPS, or FTP Requests Using an External Server

This section tells how to enable HTTP, HTTPS, or FTP filtering for inside users, and contains the following topics:

Filtering Overview

Although you can use ACLs to prevent outbound access to specific websites or FTP servers, configuring and managing web usage this way is not practical because of the size and dynamic nature of the Internet. We recommend that you use the Firewall Services Module (FWSM) in conjunction with a separate server running one of the following Internet filtering products:

- Websense Enterprise—http://www.websense.com. Supports HTTP, HTTPS, and FTP filtering.
- Sentian by N2H2—http://www.n2h2.com. Supports HTTP filtering. Although some versions of Sentian support HTTPS, the FWSM only supports HTTP with Sentian.

Because URL filtering is handled on a separate platform, the performance of the FWSM is less affected. However, filtering can considerably increase access times to websites or FTP servers when the filtering server is remote from the FWSM.

When a user issues an HTTP, HTTPS, or FTP GET request, the FWSM sends the request to the web/FTP server as well as to the filtering server at the same time. If the filtering server permits the connection for the user, then the following action occurs for each request type:

- For HTTP, the FWSM allows the reply from the web server to reach the user who issued the original request.
- For HTTPS, the FWSM allows the completion of SSL connection negotiation, and allows the reply from the web server to reach the user who issued the original request.
- For FTP, the FWSM allows the successful FTP return code to reach the user unchanged. For example, a successful return code is “250: CWD command successful.”

If the filtering server denies the connection, then the following action occurs for each request type:

- For HTTP, the FWSM redirects the user to a block page, indicating that access was denied.
- For HTTPS, the FWSM prevents the completion of SSL connection negotiation. The browser displays an error message such as "The Page or the content cannot be displayed."
- For FTP, the FWSM alters the FTP return code to show that the connection was denied. For example, the FWSM changes code 250 to "code 550: Directory not found."

For N2H2, if you enabled user authentication on the FWSM for HTTP, HTTPS, or FTP, then the FWSM also sends the username to the filtering server. The filtering server can then use user-specific filtering settings or provide enhanced reporting per user. See the "Configuring Authentication for Network Access" section on page 12-20 to configure user authentication. Websense supports filtering by IP address only.

Filtering applies only for outbound connections (from a higher security interface to a lower security interface) or between same security interfaces.

Configuring General Filtering Parameters

This section describes how to configure the FWSM to communicate with the filtering server and how to handle requests when the filtering server is down, how to handle long URLs, and whether to cache server addresses. This section includes the following topics:

Identifying the Filtering Server

You can identify up to four filtering servers per context. The FWSM uses the servers in order until a server responds. You can only configure one type of server (Websense or N2H2) in your configuration.

Note You must add the filtering server before you can configure filtering for HTTP or HTTPS with the **filter** command. If you remove the filtering servers from the configuration, then all **filter** commands are also removed.

To identify the filtering server(s), enter one of the following commands for each server you want to identify. Only one type of server is allowed in your configuration.

- To identify a Websense Enterprise server, enter the following command:

```
FWSM/contexta(config)# url-server (if_name) vendor websense host ip_address
[timeout seconds] [protocol tcp [version {1 | 4}] | udp]
```

  See the following options:

  - (*if_name)*—The interface through which the FWSM communicates with the server.
  - *ip_address*—The Websense server IP address.

  - **timeout** *seconds*—The number of seconds between 10 and 120 before the FWSM stops trying to connect to the server, and attempts to connect to the next server in the list (if available). The default is 30 seconds.
  - **protocol tcp** [**version** {**1** | **4**}]—Specifies that communication between the FWSM and the Websense server uses TCP, which is the default protocol. We recommend version 4, although version 1 is the default. Version 4 allows the FWSM to send authenticated usernames to the Websense server and to support URL caching.
  - **protocol udp**—Specifies UDP, which has greater throughput, but which does not support long URLs.

- To identify an N2H2 Sentian server, enter the following command:

```
FWSM/contexta(config)# url-server (if_name) vendor n2h2 host ip_address [port number]
[timeout <seconds>] [protocol {tcp | udp}]
```

  See the following options:

  - (*if_name)*—The interface through which the FWSM communicates with the server.
  - *ip_address*—The N2H2 server IP address.
  - **port** *number*—The port used to communicate with the N2H2 server. The default is 4005 for TCP or UDP. Change this value if you change the port on the N2H2 server.
  - **timeout** *seconds*—The number of seconds between 10 and 120 before the FWSM stops trying to connect to the server, and attempts to connect to the next server in the list (if available). The default is 30 seconds.
  - **protocol** {**tcp** | **udp**}—Specifies the protocol used for communication between the FWSM and the N2H2 server. TCP is the default protocol, and is recommended.

For example, to identify redundant Sentian servers, enter:

```
FWSM/contexta(config)# url-server (perimeter) vendor n2h2 host 10.0.1.1
FWSM/contexta(config)# url-server (perimeter) vendor n2h2 host 10.0.1.2
```

Buffering Replies

By default, when a user issues a request to connect to a website or FTP server, the FWSM sends the request to the web/FTP server and to the filtering server at the same time. If the filtering server does not respond before the web/FTP server, the reply from the web/FTP server is dropped.

To avoid dropping traffic, you can configure the FWSM to buffer replies from web and FTP servers. When the filtering server eventually responds, the FWSM can allow the connection.

To enable buffering, enter the following command:

```
FWSM/contexta(config)# url-block block block-buffer-limit
```

The *block-buffer-limit* sets the amount of memory assigned to the buffer from 0 to 128 blocks. Each block is 1550 bytes.

Setting the Maximum Length of Long HTTP URLs

Websense only

By default, the FWSM considers an HTTP URL to be a long URL if it is greater than 1159 characters. If the URL exceeds the maximum size, then it is dropped by default. You can set the FWSM to truncate or block a long URL when you configure HTTP filtering. (See the "Filtering HTTP URLs" section on page 14-5.)

To increase the maximum length and to set the amount of memory used for long URLs, follow these steps:

Step 1 To change the limit for long URLs from 1159 bytes (characters), enter the following command:

```
FWSM/contexta(config)# url-block url-size long-url-size
```

Enter **2**, **3**, or **4** to change the limit to 2, 3, or 4 KB.

Step 2 To set the maximum memory available for buffering long URLs, enter the following command:

```
FWSM/contexta(config)# url-block url-mempool memory-pool-size
```

The amount of memory dedicated to long URLs is limited to avoid a DoS attack, for example.

Set the size from 2 to 10240 KB. Typically, the amount of memory should be the number of sessions you want to allow times the maximum length of the URL. For example, to allow 100 sessions for 3 KB URLs, then set the memory to be 300 KB. However, we recommend setting the memory to the maximum, 10240 KB, because the FWSM has enough memory to handle the maximum number of sessions.

Caching URL Servers

After a user accesses a site, the filtering server can allow the FWSM to cache the server address for a certain amount of time, as long as every site hosted at the address is in a category that is permitted at all times. Then, when the user accesses the server again, or if another user accesses the server, the FWSM does not need to consult the filtering server again.

Note Requests for cached IP addresses are not passed to the filtering server and are not logged. As a result, this activity does not appear in any reports.

To enable caching, enter the following command:

```
FWSM/contexta(config)# url-cache {dst | src_dst} kbytes
```

See the following options:

- **dst**—Caches the destination server address for any user that accesses the server.
- **src_dst**— Caches the source and destination server address, so access is only cached for a given user at the source address.
- *kbytes*—The cache size between 1 and 128 KB.

Filtering HTTP URLs

To filter HTTP web access for specified users, or to exempt some traffic from filtering, enter the following commands:

- To identify HTTP traffic to be filtered by a filtering server, enter the following command:

  ```
  FWSM/contexta(config)# filter url [http | port[-port]] source_ip source_mask dest_ip
  dest_mask [allow] [proxy-block] [longurl-truncate | longurl-deny] [cgi-truncate]
  ```

  See the following options:

  - **http** | *port*[**-***port*]—The port to which the HTTP request is sent. **http** specifies port 80, which is commonly used, but you can specify other ports.
  - *source_ip source_mask*—The source address and mask. Specify **0 0** for all addresses. These addresses are the local, untranslated addresses. When you configure the filtering server, use these local addresses and not the translated addresses.
  - *dest_ip dest_mask*—The destination server address and mask. Specify **0 0** for all addresses. You typically specify all addresses and allow the filtering server to determine the websites that are allowed.
  - **allow**—When the filtering server is unavailable, this option allows connections to pass without filtering. Without this option, the FWSM stops HTTP traffic until the filtering server is back online.
  - **proxy-block**—Prevents users from connecting to an HTTP proxy server.
  - **longurl-truncate** | **longurl-deny**—By default, if a URL is longer than the maximum length then the FWSM drops the packet. (The default maximum length is 1159 bytes, but can be made larger for Websense. See the "Setting the Maximum Length of Long HTTP URLs" section on page 14-4). If you specify the **longurl-truncate** option, the FWSM sends the host name or IP address portion of the URL for evaluation to the filtering server. The **longurl-deny** option denies the URL, and forwards the user to the block page.
  - **cgi-truncate**—Truncates CGI URLs to include only the CGI script location and the script name (but not parameters). Many long HTTP requests are CGI requests. If the parameters list is very long, waiting and sending the complete CGI request including the parameter list can waste memory resources and impact performance.

- To exempt traffic from being filtered, enter the following command:

  ```
  FWSM/contexta(config)# filter url except source_ip source_mask dest_ip dest_mask
  ```

For example, to filter all HTTP requests from the 10.1.1.0 network to any web server, but to exempt an administrator user (10.1.1.1) from filtering, enter the following commands:

```
FWSM/contexta(config)# filter url except 10.1.1.1 255.255.255.255 0 0
FWSM/contexta(config)# filter url http 10.1.1.0 255.255.255.0 0 0 longurl-truncate
cgi-truncate
```

To filter users only on the 10.1.2.0 network, enter the following commands:

```
FWSM/contexta(config)# filter url http 10.1.2.0 255.255.255.0 0 0
```

Filtering HTTPS URLs

Websense only

To filter HTTPS web access for specified users, enter the following command:

```
FWSM/contexta(config)# filter https source_ip source_mask dest_ip dest_mask [allow]
```

HTTPS content is encrypted, so the FWSM sends the URL lookup to the filtering server without directory and filename information.

For the source addresses, specify the local, untranslated addresses. When you configure the filtering server, use these local addresses and not the translated addresses.

When the filtering server is unavailable, the **allow** keyword allows connections to pass without filtering. Without this option, the FWSM stops HTTPS traffic until the filtering server is back online.

Filtering FTP Requests

Websense only

To enable FTP filtering, enter the following command:

```
FWSM/contexta(config)# filter ftp port source_ip source_mask dest_ip dest_mask [allow]
[interact-block]
```

Websense only filters FTP GET commands and not PUT commands.

For the source addresses, specify the local, untranslated addresses. When you configure the filtering server, use these local addresses and not the translated addresses.

When the filtering server is unavailable, use the **allow** keyword allows connections to pass without filtering. Without this option, the FWSM stops FTP traffic until the filtering server is back online.

The **interactive-block** keyword prevents interactive FTP sessions that do not provide the entire directory path. An interactive FTP client is a non-browser client such as the **ftp** command from a DOS prompt or a UNIX shell prompt, or a stand alone FTP client. For example, when you use a web browser for FTP and you browse to a file, the URL for the file includes the entire path. When you use the **ftp** command at the command line, you can change directories without typing the entire path (**cd ./files** instead of **cd /public/files**), in which case the firewall cannot determine your exact location.

Viewing Filtering Statistics

This section describes how to monitor filtering statistics, and includes the following topics:

Viewing Filtering Server Statistics

To show information about the filtering server or to show statistics, enter the following command:

```
FWSM/contexta# show url-server stats
```

The following sample display shows filtering statistics:

```
FWSM/contexta# show url-server stats
URL Server Statistics:
----------------------
Vendor                           websense
URLs total/allowed/denied        50/35/15
HTTPSs total/allowed/denied      1/1/0
FTPs total/allowed/denied        3/1/2

URL Server Status:
------------------
10.130.28.18            UP

URL Packets Sent and Received Stats:
------------------------------------
Message                 Sent    Received
STATUS_REQUEST          65155   34773
LOOKUP_REQUEST          0       0
LOG_REQUEST             0       NA
------------------------------------
```

Viewing Caching Statistics

To show URL caching statistics, enter the following command:

```
FWSM/contexta# show url-cache stats
```

The following sample display shows how the cache is used:

```
FWSM/contexta# show url-cache stats
URL Filter Cache Stats
----------------------
    Size :     128KB
 Entries :      1724
  In Use :       456
 Lookups :        45
    Hits :         8
```

Viewing Filtering Performance Statistics

To show URL filtering performance statistics (as well as other performance statistics), enter the following command:

```
FWSM/contexta# show perfmon
```

The following sample display shows filtering statistics in the URL Access and URL Server Req rows:

```
FWSM/contexta# show perfmon
PERFMON STATS:    Current      Average
Xlates               0/s          0/s
Connections          0/s          2/s
TCP Conns            0/s          2/s
UDP Conns            0/s          0/s
URL Access           0/s          2/s
URL Server Req       0/s          3/s
TCP Fixup            0/s          0/s
TCPIntercept         0/s          0/s
HTTP Fixup           0/s          3/s
FTP Fixup            0/s          0/s
AAA Authen           0/s          0/s
AAA Author           0/s          0/s
AAA Account          0/s          0/s
```

CHAPTER 15

Using Failover

This chapter describes the Firewall Services Module (FWSM) failover feature, which allows a secondary FWSM to take over the functionality of a failed FWSM. This chapter includes the following sections:

Note See the "Configuring the Switch for Failover" section on page 2-11 to configure the switch for failover.

Understanding Failover

This section describes how failover works and includes the following sections:

Failover Overview

The failover feature lets you use a standby FWSM to take over the functionality of a failed FWSM. Failover is compatible with both routed and transparent firewall modes, and with single and multiple context modes.

Note The two FWSMs must have the same major (first number) and minor (second number) software version, license, and operating modes (routed or transparent, single or multiple context). You can use different maintenance versions (third numbers) during an upgrade process; for example, you can upgrade one unit from 2.2(1) to 2.2(2) and failover is still active. However, we recommend upgrading both units to the same version to ensure long-term compatibility. We do not guarantee full compatibility for failover when the maintenance versions differ.

When the active unit fails, it changes to the standby state, while the standby unit changes to the active state.

The unit that becomes active takes over the active unit IP addresses (or, for transparent firewall, the management IP address) and MAC address, and it begins passing traffic. The FWSM has one MAC address for all interfaces. The unit that was active and is now in standby state takes over the standby IP addresses and MAC address.

Because network devices see no change in the MAC to IP address pairing, failover is unnoticed by the rest of the network. However, the host switch needs to reassociate the new active and standby chassis slots with their corresponding MAC addresses. The FWSM helps this process by sending out gratuitous ARPs on all its VLAN interfaces. (See the "Primary/Secondary Status and Active/Standby Status" section on page 15-10 section for more information about MAC addresses).

The standby unit can effectively take over as the active unit because it has the same configuration, and it is assigned the same VLANs from the switch.

Note For multiple context mode, the FWSM can fail over the entire module (including all contexts) but cannot fail over individual contexts separately.

Regular and Stateful Failover

The FWSM supports two types of failover:

- Regular failover—When a failover occurs, all active connections are dropped and clients need to reestablish connections when the new active unit takes over.
- Stateful failover—During normal operation, the active unit continually passes per-connection stateful information (for each context) to the standby unit. The interval between stateful information updates is 10 seconds, but if you set the unit polltime to be greater than 10 seconds, then that interval is used.

  After a failover occurs, the same connection information is available at the new active unit. Supported end-user applications are not required to reconnect to keep the same communication session.

The state information passed to the standby unit includes the following data:

- NAT translation table
- TCP connection states
- UDP connection states (for connections lasting at least 15 seconds)
- HTTP connection states (Optional)
- H.323, SIP, and MGCP UDP media connections
- ARP table
- (Transparent firewall mode only) MAC address table

Failover and State Links

This section describes the failover link and, for stateful failover, the state link, and it includes the following topics:

- Failover Link, page 15-3
- State Link, page 15-3

Failover Link

The two units constantly communicate over a failover link to determine the operating status of each unit. Communications over the failover link include the following data:

- The unit state (active or standby).
- Hello messages (also sent on all other interfaces).
- Configuration synchronization between the two units. (See the "Configuration Replication" section on page 15-10 section for more information.)

The failover link uses a special VLAN interface that you do not configure as a normal networking interface; rather, it exists only for failover communications. This VLAN should only be used for the failover link (and optionally for the state link).

For multiple context mode, the failover link resides in the system configuration. This interface (and the state link, if used) is the only configurable interface in the system configuration.

Note The IP address and MAC address for the failover link do not change at failover.

State Link

To use stateful failover, configure a state link to pass all state information. This link can be the same as the failover link, but we recommend that you assign a separate VLAN and IP address for the state link. The state traffic can be large, and performance is improved with separate links.

In multiple context mode, the state link resides in the system configuration. This interface and the failover interface are the only interfaces in the system configuration.

Note The IP address and MAC address for the state link do not change at failover.

Module Placement

You can place the primary and secondary FWSMs within the same switch or in two separate switches. The following sections describe each option:

- Intra-Chassis Failover, page 15-4
- Inter-Chassis Failover, page 15-4

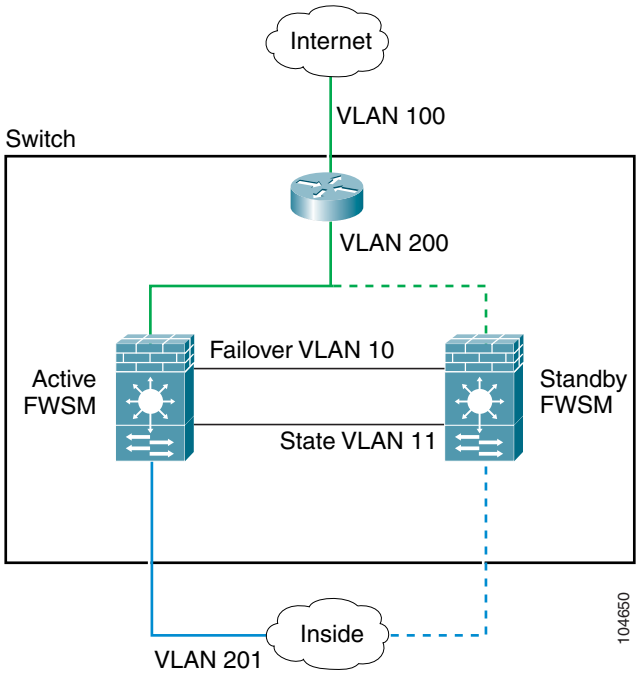
Intra-Chassis Failover

If you install the secondary FWSM in the same switch as the primary FWSM, you protect against module-level failure. To protect against switch-level failure, as well as module-level failure, see the "Inter-Chassis Failover" section on page 15-4.

Even though both FWSMs are assigned the same VLANs, only the active unit takes part in networking. The standby unit does not pass any traffic.

Figure 15-1 shows a typical intra-switch configuration.

Figure 15-1 Intra-Switch Failover



Inter-Chassis Failover

To protect against switch-level failure, you can install the secondary FWSM in a separate switch. The FWSM does not coordinate failover directly with the switch, but it works harmoniously with the switch failover operation. See the switch documentation to configure failover for the switch.

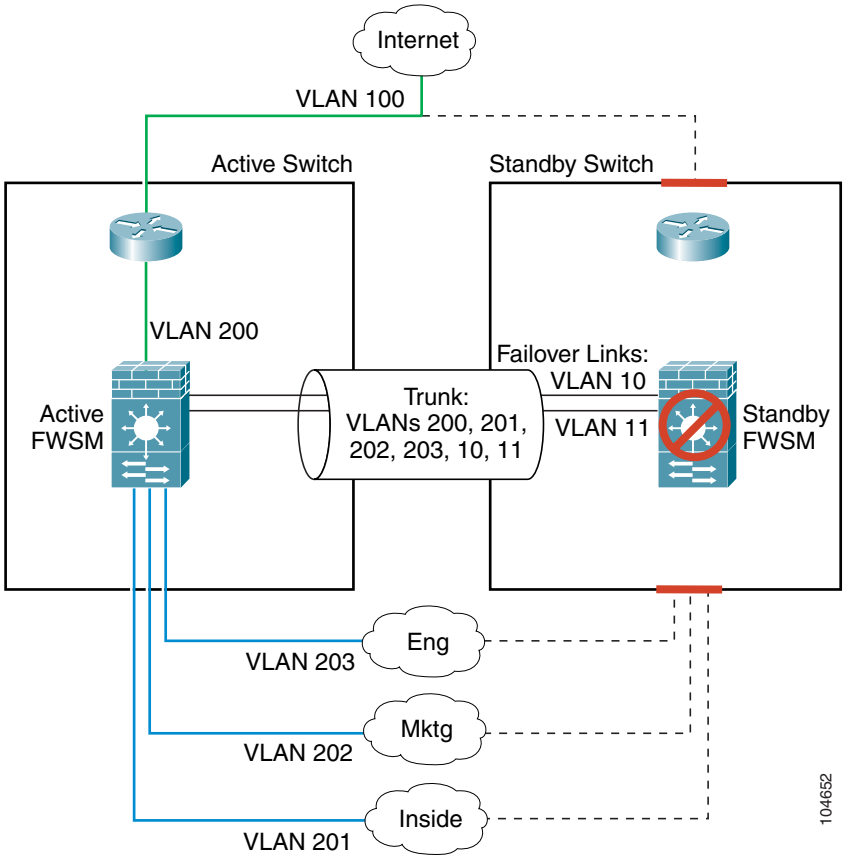
To accommodate the failover communications between the FWSMs, you must configure a trunk port between the two switches that carries all the FWSM VLANs. Because this trunk also accommodates FWSM traffic when a module fails, this trunk should be at least as large as the maximum amount of

traffic you expect to be inspected by the FWSM. The FWSM has an internal 6-Gbps EtherChannel to the switch, so if the FWSM runs at full capacity, the trunk between the two devices needs to include at least six 1-Gbps interfaces. EtherChannel aggregates the bandwidth of up to eight compatibly configured ports into a single logical link. (See the "Adding a Trunk Between a Primary Switch and Secondary Switch" section on page 2-12 for more information.)

Figure 15-2 shows a typical switch and FWSM redundancy configuration. The Spanning Tree algorithm ensures that the VLANs pass through only one switch, which also contains the active FWSM. The trunk between the two switches carries all FWSM VLANs, including the failover and state links (VLANs 10 and 11).

Note The FWSM failover is independent of the switch failover operation; however, the FWSM works in any switch failover scenario.

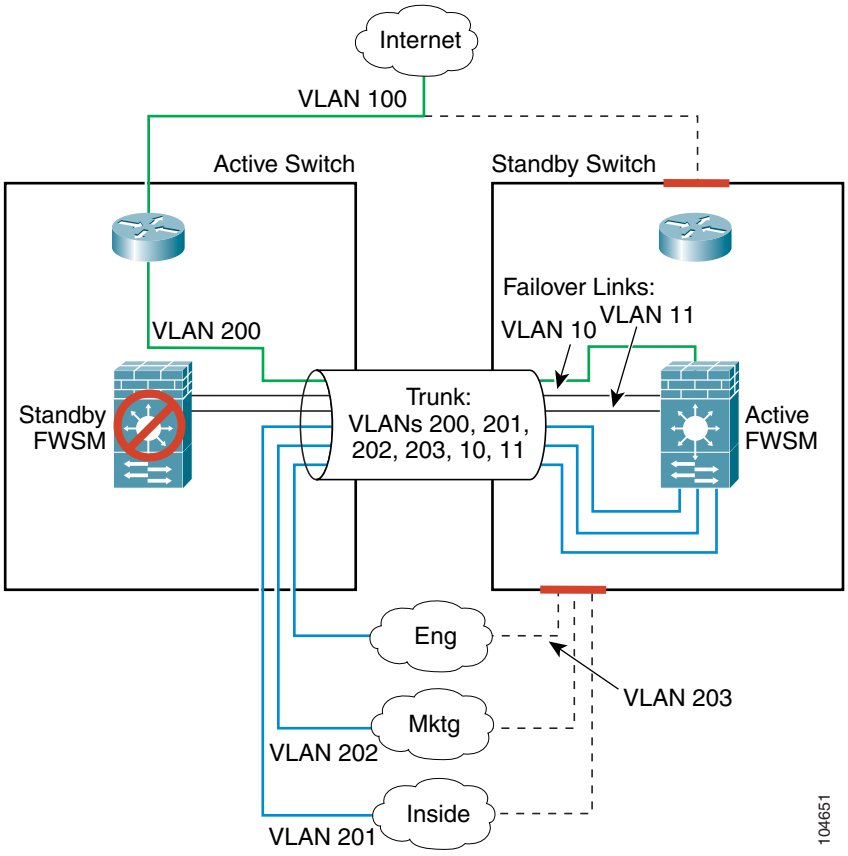
Figure 15-2 Normal Operation with Standby Units



The path that the traffic takes after failover depends on which device fails as follows:

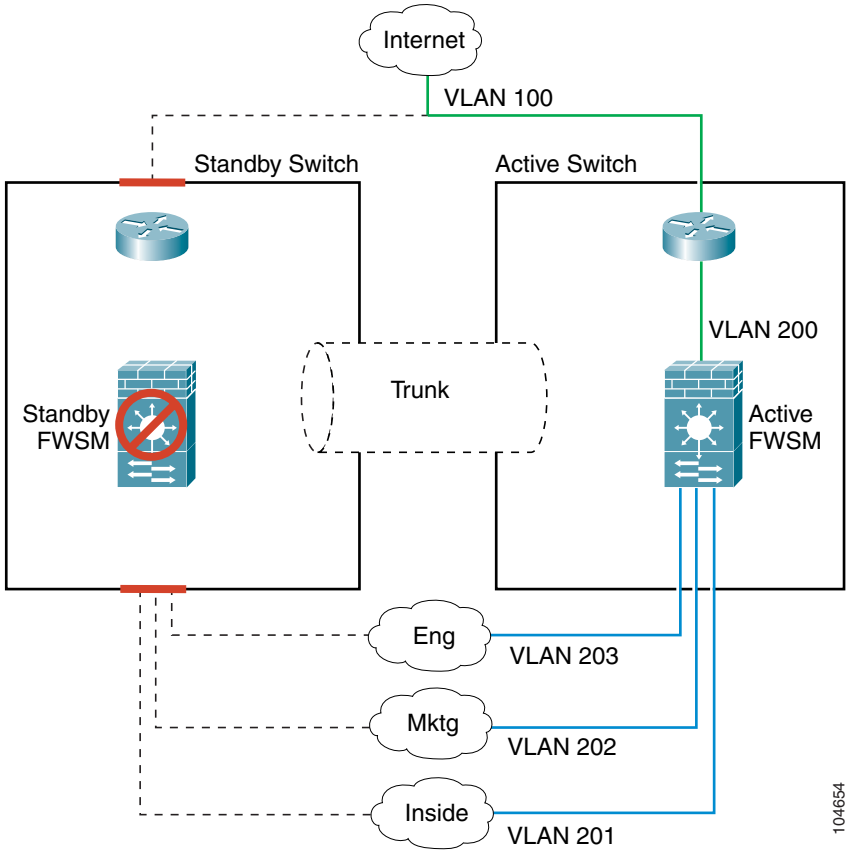
- FWSM failure only—If the primary FWSM fails, then the secondary FWSM becomes active. However, if the primary switch is still active, all VLAN traffic destined for the FWSM continues to enter the primary switch. The secondary (now active) FWSM receives and sends all traffic over the trunk (Figure 15-3).

Figure 15-3 FWSM Failure Only



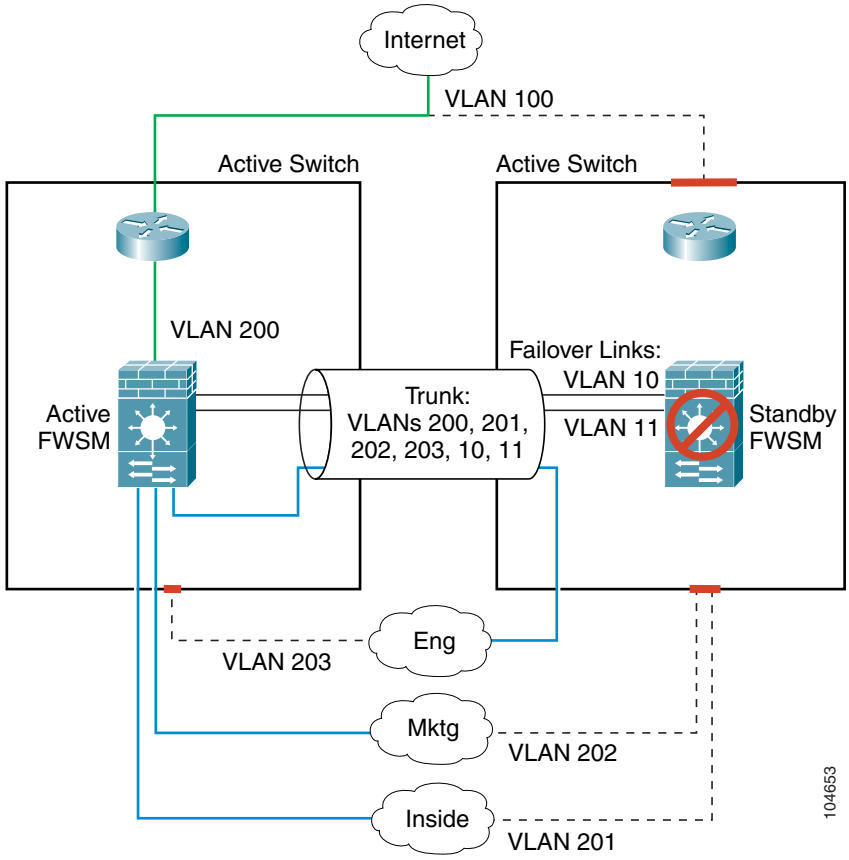
- Switch/FWSM failure—If the entire switch fails, as well as the FWSM (such as in a power failure), then both the switch and the FWSM fail over to their secondary units (Figure 15-4).

Figure 15-4 Switch/FWSM Failure



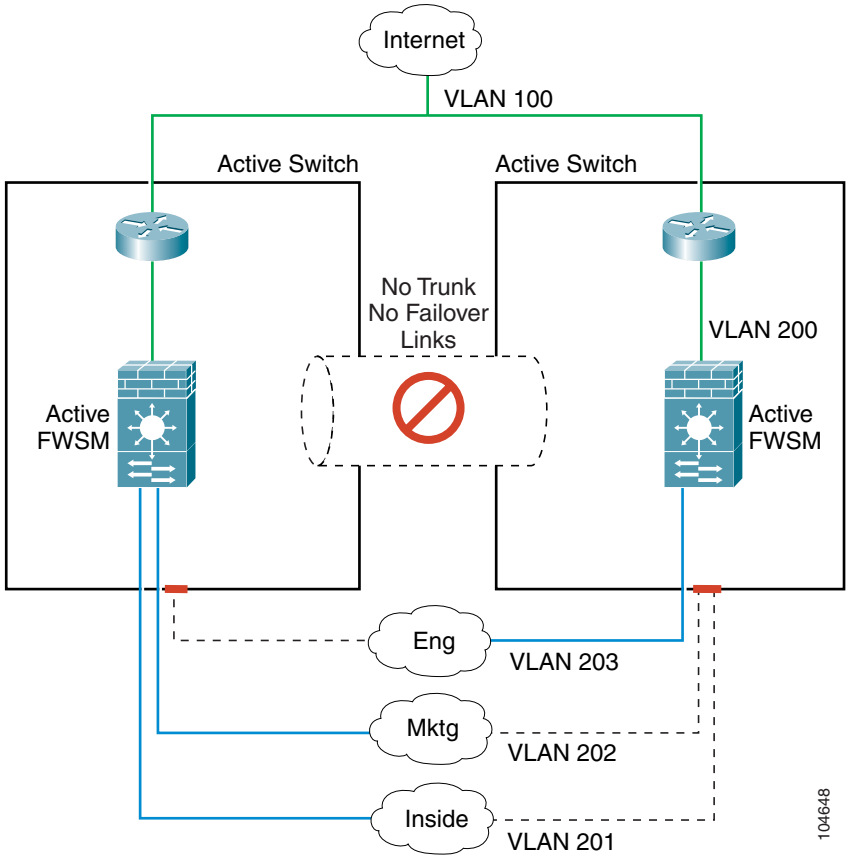
- Partial switch failure—If one or more interfaces on the switch fails, both switches would be partially active, but only one FWSM is active. The FWSM, which operates independently of the switch, has no reason to fail over because the active FWSM receives FWSM traffic from the secondary switch over the trunk (Figure 15-5).

Figure 15-5 Partial Switch Failure



- Trunk failure—If the trunk between the switches fails, all communication between the FWSMs terminates, which results in both FWSMs becoming active. Spanning Tree prevents any loops, however, and traffic is handled successfully by one or both FWSMs until you resolve the trunk issue (Figure 15-6).

Figure 15-6 Trunk Failure



Transparent Firewall Requirements

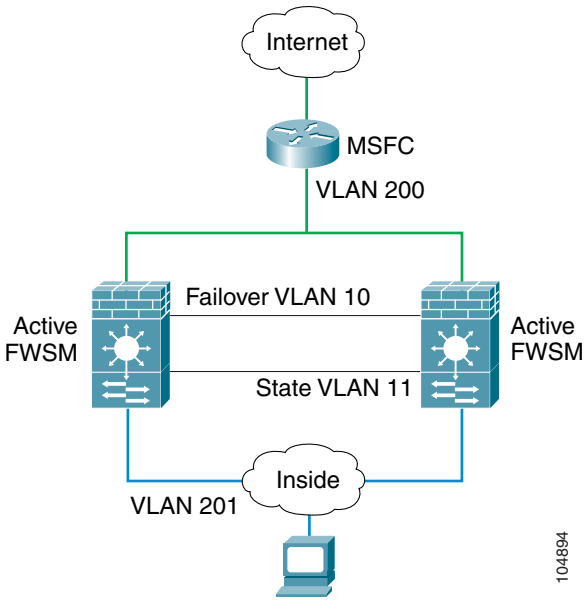
To avoid loops when you use failover in transparent mode, you must use switch software that supports BPDU forwarding, and you must configure the FWSM to allow BPDUs. See the "Chassis System Requirements" section on page 1-2 for switch software versions that allow BPDUs automatically.

To allow BPDUs through the FWSM, configure an EtherType ACL and apply it to both interfaces according to the "Adding an EtherType Access Control List" section on page 10-16.

Loops can occur if both units are active at the same time, such as when both units are discovering each other's presence, or due to a bad failover link as described in the "Basic Failover Questions" section on page 15-24. Because the FWSM units bridge packets between the same two VLANs, loops can occur

when inside packets destined for the outside get endlessly replicated by both FWSMs (see Figure 15-7). The spanning tree protocol can break such loops if there is a timely exchange of BPDUs. To break the loop, BPDUs sent between VLAN 200 and VLAN 201 need to be bridged.

Figure 15-7 Potential Loops in Transparent Mode



Primary/Secondary Status and Active/Standby Status

The main differences between the two units in a failover pair are related to which unit is active and which unit is standby, namely which IP addresses to use and which unit actively passes traffic.

However, a few differences exist between the units based on which unit is primary (as specified in the configuration) and which unit is secondary:

- The primary unit always becomes the active unit if both units start up at the same time (and are of equal operational health).
- The primary unit MAC address is always coupled with the active IP addresses. The exception to this rule occurs when the secondary unit is active, and cannot obtain the primary MAC address over the failover link. In this case, the secondary MAC address is used.

Configuration Replication

The two FWSM units share almost the identical configuration. The configuration can be the same because it includes both the active IP addresses and the standby IP addresses. When a unit is active, it uses the active IP addresses; when a unit is standby, it uses the standby IP addresses.

Note Because the configuration is the same on both units, the host names, usernames, and passwords are also the same.

The only difference in the configuration is the primary and secondary designation, although you must also pre-configure the failover link on the secondary unit before the units can communicate. All other configuration is automatically replicated from the active to the standby unit.

The active unit sends the configuration in running memory to the standby unit. On the standby unit, the configuration exists only in running memory. You can optionally save the configuration to Flash memory so that when you reboot the standby unit when the active unit is unavailable, the standby unit can become the active unit. To save the configuration to Flash memory after replication:

- For single context mode, enter the **copy running-config startup-config** command on the active unit. The command is replicated to the standby unit, which proceeds to write its configuration to Flash memory.
- For multiple context mode, enter the **copy running-config startup-config** command on the active unit from the system execution space and within each context on disk. The command is replicated to the standby unit, which proceeds to write its configuration to Flash memory. Contexts with startup configurations on external servers are accessible from either unit over the network and do not need to be saved separately for each unit. Alternatively, you can copy the contexts on disk from the active unit to an external server, and then copy them to disk on the standby unit. (See the "Downloading a Text Configuration" section on page 16-6 for more information.)

Configuration replication from the active unit to the standby unit occurs in the following circumstances:

- When the standby unit completes its initial startup, it clears its running configuration (except for the failover commands that must be pre-configured and are not replicated), and the active unit sends its entire configuration to the standby unit.
- As commands are entered on the active unit, they are sent across the failover link to the standby unit. You do not have to save the active configuration to Flash memory to replicate the commands.
- If you enter the **write standby** command on the active unit, the standby unit clears its running configuration (except for the failover commands that must be pre-configured and are not replicated), and the active unit sends its entire configuration to the standby unit.

  For multiple context mode, when you enter the **write standby** command in the system execution space, all contexts are replicated. If you enter the **write standby** command within a context, the command replicates only the context configuration.

Note Changes made on the standby unit are not replicated to the active unit. If you enter a command on the standby unit, the FWSM displays the message "**** WARNING **** Configuration Replication is NOT performed from Standby unit to Active unit. Configurations are no longer synchronized." This message displays even when you enter many commands that do not affect the configuration.

When the replication starts, the FWSM console displays the message "Beginning configuration replication: Sending to mate," and when it is complete, the FWSM displays the message "End Configuration Replication to mate." During the replication, information cannot be entered on the FWSM terminal. Depending on the size of the configuration, replication can take several minutes.

Failover Triggers

The unit can fail if one of the following events occurs:

- The unit has a hardware failure or a power failure.
- The unit has a software failure.
- Too many monitored interfaces fail.

Because the FWSM can have a large number of interfaces, it cannot monitor every interface. Rather, you configure the FWSM to monitor a subset of interfaces. The FWSM fails over when a certain number of monitored interfaces fails; you configure the failure threshold to be an absolute value or a percentage of the total number of monitored interfaces.

See the "Failover Monitoring" section on page 15-13 for more information about when a unit or interface is considered to be failed.

Failover Actions

Table 15-1 shows the failover action for each failure event.

Table 15-1 Failover Behavior

Failure Event	Policy	Active Action	Standby Action	Notes
Active unit failed (power or hardware)	Failover	n/a	Become active Mark active as failed	No hello messages are received on any monitored interface or the failover link.
Formerly active unit recovers	No failover	Become standby	No action	None.
Standby unit failed (power or hardware)	No failover	Mark standby as failed	n/a	When the standby unit is marked as failed, then the active unit will not attempt to fail over, even if the interface failure threshold is surpassed.
Failover link failed during operation	No failover	Mark failover interface as failed	Mark failover interface as failed	You should restore the failover link as soon as possible because the unit cannot fail over to the standby unit while the failover link is down.
Failover link failed at startup	No failover	Mark failover interface as failed	Become active	If the failover link is down at startup, both units will become active.
State link failed	No failover	No action	No action	State information will become out of date, and sessions will be terminated if a failover occurs.
Interface failure on active unit above threshold	Failover	Mark active as failed	Become active	None.
Interface failure on standby unit above threshold	No failover	No action	Mark standby as failed	When the standby unit is marked as failed, then the active unit will not attempt to fail over even if the interface failure threshold is surpassed.
Trunk failure in inter-switch setup	No failover	No action	Become active	Both units become active if all communication between the modules is terminated, as in the case of a trunk failure. Neither unit receives hello messages, and both units assume the active role.

Failover Monitoring

The FWSM monitors each unit for overall health and for interface health. See the following sections for more information about how the FWSM performs tests to determine the state of each unit:

- Unit Health Monitoring, page 15-13
- Interface Monitoring, page 15-13

Unit Health Monitoring

The FWSM determines the health of the other unit by monitoring the failover link. When a unit does not receive hello messages on the failover link, then the unit sends an ARP request on all interfaces, including the failover interface. The FWSM retries a user-configurable number of times. The action the FWSM takes depends on the response from the other unit. See the following possible actions:

- If the FWSM receives a response on any interface, then it does not fail over.
- If the FWSM does not receive a response on any interface, then the standby unit switches to active mode and classifies the other unit as failed.
- If the FWSM does not receive a response on the failover link only, then the unit does not failover. The failover link is marked as failed. You should restore the failover link as soon as possible because the unit cannot fail over to the standby while the failover link is down.

Note If a failed unit does not recover and you believe it should not be failed, you can reset the state by entering the **failover reset** command. If the failover condition persists, however, the unit will fail again.

Interface Monitoring

You can monitor up to 250 interfaces divided between all contexts. You should monitor important interfaces, for example, you might configure one context to monitor a shared VLAN (because the interface is shared, all contexts benefit from the monitoring).

When a unit does not receive hello messages on a monitored interface, it runs the following tests:

1. Link Up/Down test—A test of the VLAN status. If the Link Up/Down test indicates that the VLAN is operational, then the FWSM performs network tests. The purpose of these tests is to generate network traffic to determine which (if either) unit has failed. At the start of each test, each unit clears its received packet count for its interfaces. At the conclusion of each test, each unit looks to see if it has received any traffic. If it has, the interface is considered operational. If one unit receives traffic for a test and the other unit does not, the unit that received no traffic is considered failed. If neither unit has received traffic, then the next test is used.
2. Network Activity test—A received network activity test. The unit counts all received packets for up to 5 seconds. If any packets are received at any time during this interval, the interface is considered operational and testing stops. If no traffic is received, the ARP test begins.
3. ARP test—A reading of the unit ARP cache for the 2 most recently acquired entries. One at a time, the unit sends ARP requests to these machines, attempting to stimulate network traffic. After each request, the unit counts all received traffic for up to 5 seconds. If traffic is received, the interface is considered operational. If no traffic is received, an ARP request is sent to the next machine. If at the end of the list no traffic has been received, the ping test begins.
4. Broadcast Ping test—A ping test that consists of sending out a broadcast ping request. The unit then counts all received packets for up to 5 seconds. If any packets are received at any time during this interval, the interface is considered operational and testing stops.

If all network tests fail for an interface, but this interface on the other unit continues to successfully pass traffic, then the interface is considered to be failed. If the threshold for failed interfaces is met, then a failover occurs. If the other unit interface also fails all the network tests, then both interfaces go into the "Unknown" state and do not count towards the failover limit.

An interface becomes operational again if it receives any traffic. A failed FWSM returns to standby mode if the interface failure threshold is no longer met.

Note If a failed unit does not recover and you believe it should not be failed, you can reset the state by entering the **failover reset** command. If the failover condition persists, however, the unit will fail again.

Configuring Failover

This section describes how to configure failover and includes the following topics:

- Configuring the Primary Unit, page 15-14
- Configuring the Secondary Unit, page 15-17

Configuring the Primary Unit

Follow these steps to configure the primary unit. For multiple context mode, all steps are performed in the system execution space unless otherwise noted.

Note At any time during the procedure, you can enter the **show failover** command to see the failover status. See the "Using the Show Failover Command" section on page 15-18 section for detailed information.

Step 1 To configure the VLAN interface you are using for the failover link, enter the following command. For multiple context mode, enter this command in the system execution space:

```
primary(config)# failover lan interface interface_name vlan vlan
```

Note this setting because this command is the same on the secondary unit.

This VLAN should not be used for any other purpose (except, optionally, the state link) or be assigned to any switch ports. This VLAN does need to be assigned to the FWSM by the switch.

Do not assign an ACL to this interface; failover traffic is allowed automatically, and other traffic is denied.

Step 2 To set the IP address of the failover interface, enter the following command:

```
primary(config)# failover interface ip failover_interface ip_address mask standby
ip_address
```

The standby IP address must be in the same subnet as the active IP address. You do not need to identify the standby address subnet mask.

Note this setting because this command is the same on the secondary unit.

The failover link IP address and MAC address do not change at failover. The active IP address always stays with the primary unit, while the standby IP address stays with the secondary unit.

Step 3 (Stateful failover only) To configure the VLAN interface you are using for the state link, enter the following command:

```
primary(config)# failover link interface_name [vlan vlan]
```

This VLAN should not be used for any other purpose (except, optionally, the failover link) or be assigned to any switch ports. This VLAN does need to be assigned to the FWSM by the switch.

If the interface is the same as the failover interface, you do not need to identify the VLAN.

Do not assign an ACL to this interface; failover traffic is allowed automatically, and other traffic is denied.

Step 4 (Stateful failover only) To set the IP address of the state interface, enter the following command:

```
primary(config)# failover interface ip state_interface ip_address mask standby ip_address
```

The standby IP address must be in the same subnet as the active IP address. You do not need to identify the standby address subnet mask.

The state link IP address and MAC address do not change at failover. The active IP address always stays with the primary unit, while the standby IP address stays with the secondary unit.

Step 5 (Stateful failover only—Optional), To allow HTTP connections to be included in the state information, enter the following command:

```
primary(config)# failover replication http
```

If you do not allow HTTP replication, then HTTP connections are disconnected at failover. HTTP connections are brief and frequent, and the state information, although updated constantly, might not include the latest HTTP states at failover. For this reason, you might want to disable HTTP replication to reduce the amount of traffic on the state link.

Step 6 To set the threshold for monitored interface failure, enter the following command:

```
primary(config)# failover interface-policy number[%]
```

When the number of failed monitored interfaces meets the value you set with this command, then the FWSM fails over. You can set the following arguments:

- *number*—An absolute value.
- *number%*—A percentage of all monitored interfaces.

Step 7 To set this FWSM as the primary unit, enter the following command:

```
primary(config)# failover lan unit primary
```

Note This command is the only configuration difference between the primary and secondary units, although you need to set other **failover** commands on the secondary unit before the FWSM can replicate the active configuration.

Step 8 (Optional) To set how often hello messages are sent on the failover link and how long to wait before testing the peer for failure if no hello messages are received, enter the following command:

```
primary(config)# failover polltime [unit] [msec] number [holdtime seconds]
```

See the following arguments:

- **polltime unit** [**msec**] *number*—The amount of time between hello messages. Set the time in seconds between 1 and 15. The default is 1 second. If you specify **msec**, you can set the time between 500 and 999 milliseconds.
- **holdtime** *number*—Sets the time during which a unit must receive a hello message on the failover link, or else the unit begins the testing process for peer failure. Set the time in seconds between 15 and 45. The default is the greater of 15 seconds or 3 times the polltime. You cannot enter a value that is less than 3 times the polltime.

  For example, if the polltime is 1 second, then a 15 second holdtime means 15 hello messages are missed before the unit is tested for failure.

Note The interval between stateful information updates is 10 seconds, but if you set the polltime to be greater than 10, then that interval is used.

Step 9 (Optional) To set the time in seconds between hello messages on monitored interfaces, enter the following command:

```
primary(config)# failover polltime interface seconds
```

If the interface does not receive five consecutive hello messages, the FWSM begins the testing process for interface failure. See the "Failover Monitoring" section on page 15-13 for more information.

The *seconds* is an integer between 3 and 15. The default is 15 seconds, which means an interface receives no reply for 75 seconds (5 times the polling interval) before the interface is tested for failure.

Step 10 To enable failover, enter the following command:

```
primary(config)# failover
```

Step 11 (Multiple context mode only) To save the system configuration to Flash memory, enter the following command:

```
primary(config)# copy running-config startup-config
```

Step 12 (Multiple context mode only) To change to a context to configure the standby IP addresses (if you have not already done so) and to configure the interface monitoring, enter the following command:

```
primary(config)# changeto context name
```

Step 13 If you have not done so already, set the standby IP address for each interface (routed mode) or for the management IP address (transparent mode) by entering the command appropriate for your firewall mode.

- For routed mode, enter the following command for each interface:

  ```
  primary/contexta(config)# ip address interface_name ip_address mask standby ip_address
  ```

- For transparent mode, enter the following command:

  ```
  primary/contexta(config)# ip address ip_address mask standby ip_address
  ```

The standby IP address is used on the FWSM that is currently the standby unit.

To add the standby address, reenter the **ip address** command for each interface (or management IP address) and add the **standby** *ip_address* option.

This IP address must be in the same subnet as the active IP address. You do not identify the subnet mask. To check the current IP address settings, enter the **show ip address** command.

Step 14 To enable monitoring on an interface, enter the following command:

```
primary/contexta(config)# monitor-interface interface_name
```

The maximum number of interfaces to monitor on the FWSM (divided between all contexts) is 250.

Step 15 To save the configuration for the context (in multiple context mode) or for the single mode FWSM, enter the following command:

```
primary/contexta(config)# copy running-config startup-config
```

Step 16 (Multiple context mode only) Repeat Step 12 through Step 15 for each context.

See the "Failover Configuration Example" section on page 15-26 for a typical failover configuration.

Configuring the Secondary Unit

The only configuration required for the secondary unit is for the failover interface. The secondary unit requires these commands to initially communicate with the primary unit. After the primary unit sends its configuration to the secondary unit, the only permanent difference between the two configurations is the **failover lan unit** command, which identifies each unit as primary or secondary.

For multiple context mode, all steps are performed in the system execution space.

Note At any time during the procedure, you can enter the **show failover** command to see the failover status. See the "Using the Show Failover Command" section on page 15-18 section for detailed information.

To configure the secondary unit, follow these steps:

Step 1 If required, and if you have not already done so, enter the activation key to enable the same number of contexts as are licensed on the primary unit by entering the following command:

```
secondary(config)# activation-key key
```

Step 2 If you have not already done so, set the context mode to match the primary unit by entering the following command:

```
secondary(config)# mode {single | multiple}
```

The FWSM reboots.

Step 3 To configure the VLAN interface you are using for the failover link, enter the following command:

```
secondary(config)# failover lan interface interface_name vlan vlan
```

Use the same setting as the primary unit.

Step 4 To set the IP address of the failover interface, enter the following command:

```
secondary(config)# failover interface ip interface_name ip_address mask standby ip_address
```

Use the same setting as the primary unit.

Step 5 (Optional) To set this FWSM as the secondary unit, enter the following command:

```
secondary(config)# failover lan unit secondary
```

The default is secondary.

Note This command is the only configuration difference between the primary and secondary units.

Step 6 To enable failover, enter the following command:

```
secondary(config)# failover
```

After you enable failover, the active unit sends the configuration in running memory to the standby unit. As the configuration synchronizes, the messages "Beginning configuration replication: Sending to mate" and "End Configuration Replication to mate" appear on the active unit console.

Step 7 To save the configuration to Flash memory, enter the following command:

```
secondary(config)# copy running-config startup-config
```

See the "Failover Configuration Example" section on page 15-26 for a typical failover configuration.

Verifying the Failover Configuration

This section describes how to verify your failover configuration. This section includes the following topics:

See the "Monitoring Failover" section on page 15-23 section for other troubleshooting tools.

Using the Show Failover Command

On each unit, verify the failover status by entering the following command in the system execution space:

```
primary(config)# show failover
```

This command shows the following information:

- The failover status, either on or off
- The active unit
- The IP addresses assigned for the active and standby units
- The failover link information
- The interface policy
- The stateful failover statistics

See the following sample **show failover** command output. A description of each field follows.

```
FWSM(config)# show failover
Failover On
Failover unit Primary
Failover LAN Interface fover Vlan 150
Unit Poll frequency 15 seconds
Interface Poll frequency 15 seconds
Interface Policy 50%
Monitored Interfaces 249 of 250 maximum
Last Failover at: 10:58:08 Apr 15 2004
        This host: Primary - Active
                Active time: 2232 (sec)
                admin Interface inside (10.6.8.91): Normal
                admin Interface outside (70.1.1.2): Normal
        Other host: Secondary - Standby
                Active time: 0 (sec)
                admin Interface inside (10.6.8.100): Normal
                admin Interface outside (70.1.1.3): Normal

Stateful Failover Logical Update Statistics
        Link : 4th
        Stateful Obj    xmit       xerr       rcv        rerr
        General         0          0          0          0
        sys cmd         0          0          0          0
        up time         0          0          0          0
        xlate           0          0          0          0
        tcp conn        0          0          0          0
        udp conn        0          0          0          0
        ARP tbl         0          0          0          0
        RIP Tbl         0          0          0          0

        Logical Update Queue Information
                        Cur     Max     Total
        Recv Q:         0       0       0
        Xmit Q:         0       0       0
```

Table 15-2 describes the **show failover** output.

Table 15-2 Show Failover Display Description

Field	Options
Failover	• On • Off
Failover Unit	• Primary • Secondary
Failover LAN Interface	Shows the interface name and VLAN for the failover link: *interface_name* **vlan** *number* If you have not configured the failover interface, the display shows: `Not configured`
Unit Poll frequency	*n* seconds The number of seconds you set with the **failover poll unit** command. The default is 15 seconds.
Interface Poll frequency	*n* seconds The number of seconds you set with the **failover poll interface** command. The default is 15 seconds.

Table 15-2 Show Failover Display Description (continued)

Field	Options
Interface Policy	*n*[%] The threshold for interface failure that you set with the **failover interface-policy** command. The default is 50%.
Monitored Interfaces	*n* of 250 maximum The number of interfaces you are monitoring.
Last Failover	The last time a failover occurred.
This host: Other host:	For each host, the display shows the following information.
Primary or Secondary	• Active—The unit is in active mode. • Standby—The unit is in standby mode, • Disabled—The unit has failover disabled, or the failover link is not configured. • Listen—The unit is attempting to discover an active unit by listening for polling messages. • Learn—The unit detected an active unit, and is not synchronizing the configuration before going to standby mode. • Failed—The unit is failed.
Active time:	*n* (sec) The amount of time the unit has been in the active state. This time is cumulative, so the standby unit, if it was active in the past, will also show a value.
[*context_name*] Interface *name* (*n.n.n.n*):	For each interface, the display shows the IP address currently being used on each unit, as well as one of the following conditions: • Failed—The interface has failed. • Link Down—The interface line protocol is down. • Normal—The interface is working correctly. • No Link—The interface has been administratively shut down. • Unknown—The FWSM cannot determine the status of the interface. • (Waiting)—The interface has not yet received any polling messages from the other unit. • Testing—The interface is being tested. In multiple context mode, the context name appears before each interface.
Stateful Failover Logical Update Statistics	The following fields relate to the stateful failover feature. If the Link field shows an interface name, the stateful failover statistics are shown.

Table 15-2 Show Failover Display Description (continued)

Field	Options
Link	• *interface_name*—The interface used for the stateful failover link. • Unconfigured—You are not using stateful failover.
Stateful Obj	For each field type, the following statistics are used: • xmit—Number of transmitted packets to the other unit. • xerr—Number of errors that occurred while transmitting packets to the other unit. • rcv—Number of received packets. • rerr—Number of errors that occurred while receiving packets from the other unit.
General	Sum of all stateful objects.
sys cmd	Logical update system commands; for example, LOGIN and Stay Alive.
up time	Up time, which the active unit passes to the standby unit.
xlate	Translation information.
tcp conn	TCP connection information.
udp conn	Dynamic UDP connection information.
ARP tbl	Dynamic ARP table information.
RIP Tbl	Dynamic router table information.
Logical Update Queue Information	For each field type, the following statistics are used: • Cur—Current number of packets • Max—Maximum number of packets • Total—Total number of packets
Recv Q	The status of the receive queue.
Xmit Q	The status of the transmit queue.

Viewing Monitored Interfaces

To view the status of monitored interfaces, (from within the context) enter the following command:

```
primary/contexta(config)# show monitor-interface
```

For example:

```
primary/contexta(config)# show monitor-interface
        This host: Primary - Active
                Interface outside (88.1.1.2): Normal
                Interface inside (10.6.8.91): Normal
        Other host: Secondary - Standby
                Interface outside (88.1.1.3): Normal
                Interface inside (10.6.8.100): Normal
```

Testing the Failover Functionality

Follow these steps to ensure that failover works:

Step 1 Test that your primary (active) unit is passing traffic as expected by using FTP (for example) to send a file between hosts on different interfaces.

Step 2 Force a failover to the standby unit by entering the following command:

```
primary(config)# no failover active
```

Step 3 Use FTP to send another file between the same two hosts.

Step 4 If the test was not successful, enter the **show failover** command to check the failover status.

Step 5 When you are finished, you can leave the secondary unit as active or force the primary unit to be active again by entering the following command:

```
primary(config)# failover active
```

Forcing Failover

To force the standby unit to become active, enter one of the following commands:

- Enter this command on the active unit to failover to the standby unit:

  ```
  primary(config)# no failover active
  ```

- Enter this command on the standby unit to force it to become active:

  ```
  secondary(config)# failover active
  ```

Disabling Failover

When you disable failover, the active and standby state of each unit is maintained until you restart. For example, the standby unit remains in standby mode so that both units do not start passing traffic. To make the standby unit active (even with failover disabled), see the "Forcing Failover" section above.

To disable failover, enter the following command:

```
primary(config)# no failover
```

This command is not replicated to the standby unit so you must disable failover of the standby unit separately.

To verify that failover is off, enter the **show failover** command:

```
primary(config)# show failover
Failover Off
...
```

Monitoring Failover

When a failover occurs, both FWSMs send out system messages. This section includes the following topics:

Failover System Messages

The FWSM issues a number of system messages related to failover at priority level 2, which indicates a critical condition. To view these messages, see the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module System Messages Guide* to enable logging and to see descriptions of the system messages.

SNMP

To receive SNMP syslog traps for failover, see the "Using SNMP" section on page 17-1 for more information.

Debug Messages

To see debug messages, enter the **debug fover** command. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information.

Frequently Asked Failover Questions

This section contains frequently asked questions about the failover features and includes the following topics:

Configuration Replication Questions

See the following questions and answers for configuration replication:

- Does configuration replication save the configuration to Flash memory on the standby unit?

  No, the configuration is only in running memory.
- How can both units be configured the same without manually entering the configuration twice?

  Commands entered on the active unit are automatically replicated to the standby unit.

- What happens if I enter commands on the standby unit?

  You will see an error message telling you that the configurations are out of sync. However, the command is still applied.

  If you enter individual commands on the active unit that are replicated to the standby unit, your alterations on the standby unit are preserved.

  If you use the **write standby** command on the active unit, it will erase any new commands you entered on the standby unit.

- What happens if I enter the **copy running-config startup-config** command on the active unit?

  The **copy running-config startup-config** command is replicated to the standby unit, which proceeds to write its configuration to Flash memory.

- What happens if the configuration in Flash memory on the secondary unit differs from the configuration on the primary unit?

  After startup, the primary unit sends its configuration to the secondary unit and erases the running configuration on the secondary unit. However, the secondary unit startup configuration remains unaltered in Flash memory.

- How can I view the running configuration and the Flash memory configuration?
  - **show running**—Shows the running configuration. You can also enter **write terminal**.
  - **show config**—Shows the configuration in Flash memory.

- Are contexts that are on disk saved to disk on the standby unit?

  No, all contexts are loaded into running memory only. The startup configuration for a context continues to reside on the primary unit disk. You can copy the context configurations to the standby unit so that if the standby unit starts up and needs to be the active unit, it can load the contexts from disk.

- Can I fail over a context, but not the entire module?

  No, you can only have one active FWSM.

Basic Failover Questions

See the following questions and answers for basic failover:

- Which unit becomes active if you restart both units?

  The primary unit.

- What happens if the active unit has a power failure?

  After hello messages are not acknowledged, the standby unit becomes active.

- What happens when the formerly active unit comes online again?

  No failover occurs. It remains in standby mode.

- How long does it take to detect a failure?
  - Network errors are detected within three consecutive polling intervals (by default, 15 second intervals). The polling interval is user-configurable using the **failover poll interface** command.
  - Failover communication errors are detected within a user-configurable number of seconds (the default is 15). The polling time is user-configurable using the **failover poll unit** command.

- What maintenance is required?

  Syslog messages are generated when any errors or switches occur. Evaluate the failed unit and fix or replace it.

- Is it possible to have both FWSM units become active at the same time?

  Yes, in the following circumstances:

  - Both units have configurations in Flash memory
  - Both units have failover enabled
  - The failover link is down at startup

  or

  In an inter-switch failover scenario, the trunk between the switches fails

- What prevents the standby unit from passing traffic?

  The FWSM failover feature ensures that only traffic aimed to the standby unit (hello messages, Telnet if enabled) is successful, while traffic aimed through the unit is dropped.

Stateful Failover Questions

See the following questions and answers for stateful failover:

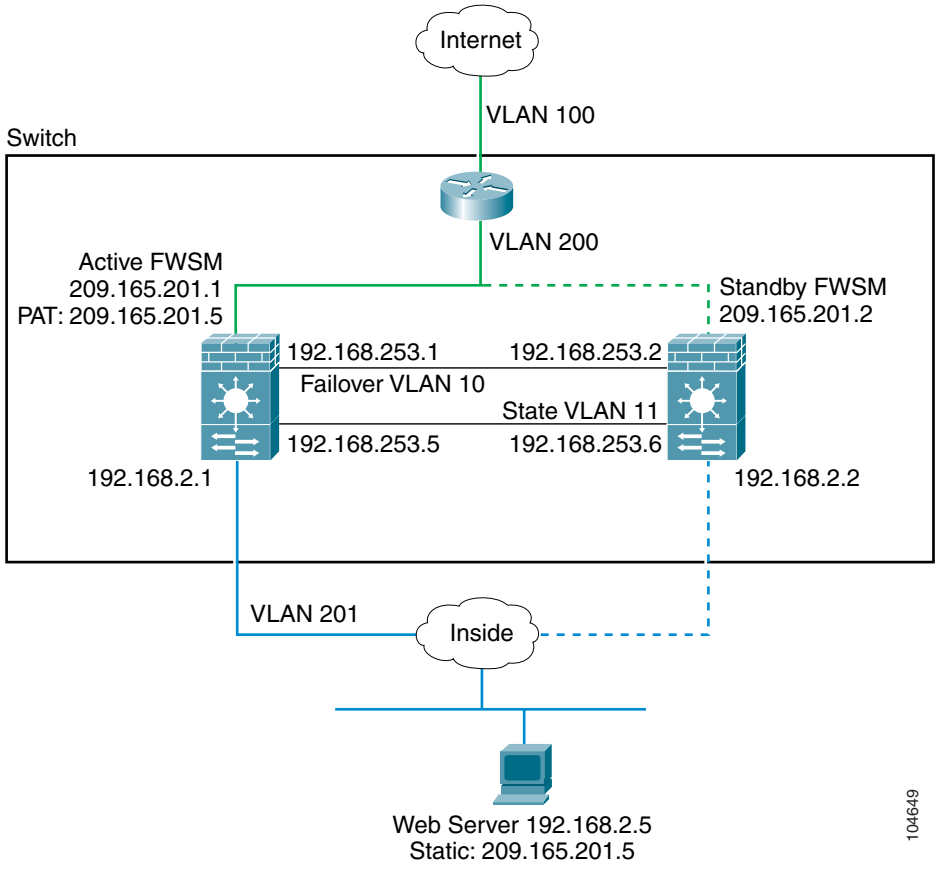
- What information is not replicated to the standby FWSM on stateful failover?
  - The user authentication (uauth) table.
  - The ISAKMP and IPSec SA table (for management access only).
  - The ARP table.
  - Routing information.
  - Other UDP connections.
- Can I share the state link interface with the failover link?

  Yes, however, we recommend that you use a separate interface.

Failover Configuration Example

Figure 15-8 shows the network diagram for a failover configuration within a switch. The only difference between the configuration of inter-switch and intra-switch failover is on the switch; the configuration on the FWSM is the same.

Figure 15-8 Failover Scenario



Example 15-1 lists the typical commands in a failover configuration. This example shows how to configure multiple context mode and shows one context, the admin context. For single context mode, simply combine the two configurations, and remove the **admin-context** command and the **context** commands.

Example 15-1 Failover Configuration

System Configuration:

```
hostname FWSM
enable password farscape
password crichton
admin-context adminctxt
context adminctxt
    allocate-interface vlan200
    allocate-interface vlan201
    config-url disk:/adminctxt.cfg
failover lan interface faillink vlan 10
failover link statelink vlan 11
failover lan unit primary
failover interface ip faillink 192.168.253.1 255.255.255.252 standby 192.168.253.2
failover interface ip statelink 192.168.253.5 255.255.255.252 standby 192.168.253.6
failover interface-policy 1
failover replication http
failover
```

Context Configuration:

```
nameif vlan200 outside security0
nameif vlan201 inside security100
enable password aeryn
password rygel
telnet 192.168.2.45 255.255.255.255 [A host on the context network, not shown]
ip address outside 209.165.201.1 255.255.255.224 standby 209.165.201.2
ip address inside 192.168.2.1 255.255.255.0 standby 192.168.2.2
monitor-interface inside
monitor-interface outside
global (outside) 1 209.165.201.3 netmask 255.255.255.224
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
static (inside,outside) 209.165.201.5 192.168.2.5 netmask 255.255.255.255 0 0
access-list acl_out permit tcp any 209.165.201.5 eq 80
access-group acl_out in interface outside
route outside 0 0 209.165.201.4 1
```

Example 15-2 shows the configuration for the secondary unit.

Example 15-2 Failover Configuration: Secondary Unit

```
failover lan interface faillink vlan 10
failover lan unit secondary
failover interface ip faillink 192.168.253.1 255.255.255.252 standby 192.168.253.2
failover
```

CHAPTER 16

Managing Software and Configuration Files

This chapter describes how to install new software on the Firewall Services Module (FWSM) from an FTP, TFTP, HTTP, or HTTPS server. You can upgrade the application software, the maintenance software, and PDM for FWSM management software.

Note If you are upgrading from a pervious release, for example Release 1.1, refer to the FWSM documentation for your version.

This chapter also describes how to download or back up a configuration file.

This chapter contains the following sections:

Installing Application or PDM Software

This section contains the following topics:

Installation Overview

To upgrade PDM, you can only install to the current application partition. For application software, you can use one of two methods to upgrade:

- Installing to the current application partition

  The benefit of this method is you do not have to boot into the maintenance partition; instead you log in as usual and copy the new software. The activation key is maintained with this method.

  This method supports downloading from a TFTP, FTP, HTTP, or HTTPS server.

  You cannot copy software to the other application partition. You might want to copy to the other partition if you want to keep the old version of software as a backup in the current partition.

You must have an operational configuration with network access. For multiple context mode, you need to have network connectivity through the admin context.

- Installing to any application partition from the maintenance partition

  The benefit of this method is you can copy software to both application partitions, and you do not have to have an operational configuration. You just need to configure some routing parameters in the maintenance partition so you can reach the server on VLAN 1.

  The disadvantage is that you need to boot into the maintenance partition, which might not be convenient if you have an operational application partition. Also, if you are running maintenance software Version 1.1, the activation key, if present, is removed and the mode reverts to single context mode. We suggest that you upgrade the maintenance software to Version 2.1 or later to keep the activation key and mode. See the "Installing Maintenance Software" section on page 16-5 to upgrade. To view the maintenance software version, log into the maintenance partition (see the "Installing Application Software to Any Application Partition" section on page 16-3), and enter **show version**.

  This method supports downloading from an FTP server only.

See the "Managing the Firewall Services Module Boot Partitions" section on page 2-12 for more information about application and maintenance partitions.

Installing Application or PDM Software to the Current Partition

When you log into the FWSM during normal operation, you can copy the application software or PDM software to the current application partition from a TFTP, FTP, HTTP, or HTTPS server.

For multiple context mode, you must be in the system execution space.

Make sure you have network access to the server:

- For single context mode, configure any interface, its IP address, and any static routes required to reach the server. See the "Configuring Interfaces" section on page 6-6 and then Chapter 8, "Configuring IP Addresses, Routing, and DHCP."
- For multiple context mode, you must first add the admin context and configure interfaces, IP addresses, and routing to provide network access. See the "Configuring a Security Context" section on page 5-17, and then the "Configuring Interfaces" section on page 6-6 and Chapter 8, "Configuring IP Addresses, Routing, and DHCP."

To copy the application or PDM software, enter one of the following commands for the appropriate download server:

- To copy from a TFTP server, enter the following command:

  ```
  FWSM# copy tftp://server[/path]/filename flash:[image | pdm]
  ```

  The **image** keyword (default) copies the application software, and the **pdm** keyword copies the PDM software.

- To copy from an FTP server, enter the following command:

  ```
  FWSM# copy ftp://[user[:password]@]server[/path]/filename[;type=xx]
  flash:[image | pdm]
  ```

  The **image** option (default) copies the application software, and the **pdm** option copies the PDM software.

  The **type** can be one of the following keywords:

  - **ap**—ASCII passive mode

  - **an**—ASCII normal mode
  - **ip**—(Default) Binary passive mode
  - **in**—Binary normal mode

  Use binary for image files.

- To copy from an HTTP or HTTPS server, enter the following command:

  ```
  FWSM# copy http[s]://
  [user[:password]@]server[:port][/path]/filename flash:[image | pdm]
  ```

  The **image** option (default) copies the application software, and the **pdm** option copies the PDM software.

For example, to copy the application software from a TFTP server, enter:

```
FWSM# copy tftp://209.165.200.226/cisco/c6svc-fwm-k9.2-1-1.bin flash:image
```

To copy the application software from an FTP server, enter:

```
FWSM# copy ftp://admin:letmein@209.165.200.227/cisco/c6svc-fwm-k9.2-1-1.bin;type=ip
flash:image
```

To copy PDM from an HTTPS server, enter:

```
FWSM# copy http://admin:letmein@209.165.200.228/pdm/pdm-411.bin flash:pdm
```

Installing Application Software to Any Application Partition

If you log into the maintenance partition, you can install application software to either application partition (cf:4 or cf:5).

Note The FWSM maintenance partition can only use VLAN 1 on the switch. The FWSM does not support 802.1Q tagging on VLAN 1.

If you are running maintenance software release 1.1, the activation key, if present, is removed and the mode reverts to single context mode. We suggest that you upgrade the maintenance software to Release 2.1 or later to keep the activation key and mode. See the "Installing Maintenance Software" section on page 16-5 to upgrade. To view the maintenance software version, log into the maintenance partition (see the "Installing Application Software to Any Application Partition" section on page 16-3), and enter **show version**.

To install application software from an FTP server while logged into the maintenance partition, follow these steps:

Step 1 To boot the FWSM into the maintenance partition, enter the command for your operating system:

- For Cisco IOS software, enter the following command:

  ```
  Router# hw-module module mod_num reset cf:1
  ```

- For Catalyst OS, enter the following command:

  ```
  Console> (enable) reset mod_num boot cf:1
  ```

Step 2 To session into the FWSM, enter the command for your operating system:

- For Cisco IOS software, enter the following command:

```
Router# session slot mod_num processor 1
```

- For Catalyst OS, enter the following command:

```
Console> (enable) session mod_num
```

Step 3 To log into the FWSM maintenance partition as root, enter the following command:

```
Login: root
```

Step 4 Enter the password at the prompt:

```
Password:
```

By default, the password is "cisco."

Step 5 To assign an IP address to the maintenance partition, enter the following command:

```
root@localhost# ip address ip _address netmask
```

This address is the address for VLAN 1, which is the only VLAN used by the maintenance partition.

Step 6 To assign a default gateway to the maintenance partition, enter the following command:

```
root@localhost# ip gateway ip_address
```

Step 7 Optional) To ping the FTP server to verify connectivity, enter the following command:

```
root@localhost# ping ftp_address
```

Step 8 To download the application software from the FTP server, enter the following command:

```
root@localhost# upgrade ftp://[user[:password]@]server[/path]/filename cf:{4 | 5}
```

cf:4 and **cf:5** are the application partitions on the FWSM.

Follow the screen prompts during the upgrade.

The configuration file in the application partition is backed up and restored at the end of the upgrade operation.

Step 9 To log out of the maintenance partition, enter the following command:

```
root@localhost# logout
```

Step 10 To reboot the module into the application partition, **cf:4** or **cf:5**, enter the command for your operating system:

- For Cisco IOS, enter the following command:

```
Router# hw-module module mod_num reset cf:{4 | 5}
```

- For Catalyst OS, enter the following command:

```
Console> (enable) reset mod_num boot cf:{4 | 5}
```

Installing Maintenance Software

You can download the maintenance software from a TFTP, HTTP, or HTTPS server when you are logged into the application partition. Passwords for the root and guest accounts of the maintenance partition are retained after the upgrade.

For multiple context mode, you must be in the system execution space.

Make sure you have network access to the server:

- For single context mode, configure any interface, its IP address, and any static routes required to reach the server. See the "Configuring Interfaces" section on page 6-6 and then Chapter 8, "Configuring IP Addresses, Routing, and DHCP."
- For multiple context mode, you must first add the admin context and configure interfaces, IP addresses, and routing to provide network access. See the "Configuring a Security Context" section on page 5-17, and then the "Configuring Interfaces" section on page 6-6 and Chapter 8, "Configuring IP Addresses, Routing, and DHCP."

To upgrade the maintenance partition software, enter one of the following commands for the appropriate download server:

- To download the maintenance software from a TFTP server, enter the following command:

  ```
  FWSM# upgrade-mp tftp[://server[:port][/path]/filename]
  ```

  If you do not enter the TFTP server information, you are prompted for the server information.

- To download the maintenance software from an HTTP or HTTPS server, enter the following command:

  ```
  FWSM# upgrade-mp http[s]://[user[:password]@]server[:port][/path]/filename
  ```

The following example shows the prompts for the TFTP server information:

```
FWSM# upgrade-mp tftp
Address or name of remote host [127.0.0.1]? 10.1.1.5
Source file name [cdisk]? mp.2-1-0-3.bin.gz
copying tftp://10.1.1.5/mp.2-1-0-3.bin.gz to flash
[yes|no|again]? yes
!!!!!!!!!!!!!!!!!!!!!!
Received 1695744 bytes.
Maintenance partition upgraded.
```

Downloading and Backing Up Configuration Files

This section describes how to download and back up configuration files, and includes the following sections:

Downloading a Text Configuration

You can download a text file from the following server types:

- TFTP
- FTP
- HTTP
- HTTPS

Make sure you have network access to the server:

- For single context mode, configure any interface, its IP address, and any static routes required to reach the server. See the "Configuring Interfaces" section on page 6-6 and then Chapter 8, "Configuring IP Addresses, Routing, and DHCP."
- For multiple context mode, add the admin context and configure interfaces, IP addresses, and routing to provide network access. See the "Configuring a Security Context" section on page 5-17, and then the "Configuring Interfaces" section on page 6-6 and Chapter 8, "Configuring IP Addresses, Routing, and DHCP."

To download a text configuration from a server, follow these steps:

Step 1 To copy the single mode startup configuration or the multiple mode system startup configuration from the server to Flash memory, enter one of the following commands for the appropriate download server:

- To copy from a TFTP server, enter the following command:

  ```
  FWSM# copy tftp://server[/path]/filename startup-config
  ```

- To copy from an FTP server, enter the following command:

  ```
  FWSM# copy ftp://[user[:password]@]server[/path]/filename[;type=xx] startup-config
  ```

  The **type** can be one of the following keywords:

  - **ap**—ASCII passive mode
  - **an**—ASCII normal mode
  - **ip**—(Default) Binary passive mode
  - **in**—Binary normal mode

  You can use ASCII or binary for configuration files.

- To copy from an HTTP or HTTPS server, enter the following command:

  ```
  FWSM# copy http[s]://[user[:password]@]server[:port][/path]/filename startup-config
  ```

For example, to copy the configuration from a TFTP server, enter the following command:

```
FWSM# copy tftp://209.165.200.226/configs/startup.cfg startup-config
```

To copy the configuration from an FTP server, enter the following command:

```
FWSM# copy ftp://admin:letmein@209.165.200.227/configs/startup.cfg;type=an startup-config
```

To copy the configuration from an HTTP server, enter the following command:

```
FWSM# copy http://209.165.200.228/configs/startup.cfg startup-config
```

Step 2 (Multiple context mode only) To copy context configurations to disk, including the admin configuration, enter one of the following commands for the appropriate download server:

- To copy from a TFTP server, enter the following command:

```
FWSM# copy tftp://server[/path]/filename disk:[path/]filename
```

- To copy from a FTP server, enter the following command:

```
FWSM# copy ftp://[user[:password]@]server[/path]/filename[;type=xx] disk:[path/]filename
```

The **type** can be one of the following keywords:

- **ap**—ASCII passive mode
- **an**—ASCII normal mode
- **ip**—(Default) Binary passive mode
- **in**—Binary normal mode

You can use ASCII or binary for configuration files.

- To copy from a HTTP or HTTPS server, enter the following command:

```
FWSM# copy http[s]://[user[:password]@]server[:port][/path]/filename disk:[path/]filename
```

Step 3 Copy the new startup configuration to the running configuration using one of these options:

- To merge the startup configuration with the current running configuration, enter the following command:

```
FWSM(config)# copy startup-config running-config
```

- To load the startup configuration and discard the running configuration, restart the FWSM by entering the following command:

```
FWSM# reboot
```

Backing Up the Configuration

To back up your configuration, copy it to an external server. Use one of the following methods:

- Copying the Configuration to a Server, page 16-7
- Copying the Configuration from the Terminal Display, page 16-8

Copying the Configuration to a Server

You can back up configuration files in the following circumstances:

- Backing up the Single Mode Configuration or Multiple Mode System Configuration, page 16-7
- Backing Up a Context Configuration within the Context, page 16-8

Backing up the Single Mode Configuration or Multiple Mode System Configuration

In single context mode, or from the system configuration in multiple mode, you can copy the startup configuration, running configuration, or a configuration file by name on disk (such as the admin.cfg).

Enter one of the following commands for the appropriate backup server:

- To copy to a TFTP server, enter the following command:

```
FWSM# copy {startup-config | running-config | disk:[path/]filename}
tftp://server[/path]/filename
```

- To copy to a FTP server, enter the following command:

```
FWSM# copy {startup-config | running-config | disk:[path/]filename}
ftp://[user[:password]@]server[/path]/filename[;type=xx]
```

  The **type** can be one of the following keywords:

  - **ap**—ASCII passive mode
  - **an**—ASCII normal mode
  - **ip**—(Default) Binary passive mode
  - **in**—Binary normal mode

  Use ASCII or binary for configuration files (as in this case), and binary only for image files.

Backing Up a Context Configuration within the Context

In multiple context mode, from within a context, you can perform the following backups:

- To copy the running configuration to the startup configuration server (connected to the admin context), enter the following command:

```
FWSM/contexta# copy running-config startup-config
```

- To copy the running configuration to a TFTP server connected to the context network, follow these steps:

  a. To specify the TFTP server that is connected to the context network, enter the following command:

```
FWSM/contexta# tftp-server interface_name ip_address path[/filename]
```

  b. To copy the running configuration to the TFTP server, enter the following command:

```
FWSM/contexta(config)# write net [:filename]
```

     If you specify the filename in the **tftp-server** command (above), you do not need to identify it in the **write net** command.

  For example:

```
FWSM/contexta(config)# tftp-server 10.1.1.1 /fwsmconfigs/contextbackup.cfg
FWSM/contexta(config)# write net
```

Copying the Configuration from the Terminal Display

To print the configuration to the terminal, enter the following command:

```
FWSM# write terminal
```

Copy the output from this command, then paste the configuration into a text file.

Monitoring and Troubleshooting the Firewall Services Module

This chapter describes how to monitor and troubleshoot the Firewall Services Module (FWSM), and contains the following sections:

Monitoring the Firewall Services Module

You can monitor the FWSM using system messages or using Simple Network Management Protocol (SNMP). This section describes:

Using System Messages

The FWSM provides extensive system messages. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module System Messages Guide* to configure logging and to view system message descriptions.

Using SNMP

This section describes how to use SNMP and includes the following topics:

SNMP Overview

The FWSM provides support for network monitoring using SNMP V1. The FWSM supports traps and SNMP get requests, but does not support SNMP set requests.

You can configure the FWSM to send traps (event notifications) to a network management station (NMS), or you can use the NMS to browse the Management Information Bases (MIBs) on the FWSM. MIBs are a collection of definitions, and the FWSM maintains a database of values for each definition. Browsing a MIB entails issuing an SNMP get request from the NMS. Use CiscoWorks for Windows or any other SNMP V1, MIB-II compliant browser to receive SNMP traps and browse a MIB.

Table 17-1 lists supported MIBs and traps for the FWSM and, in multiple mode, for each context. You can download Cisco MIBs from the following website:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

After you download the MIBs, compile them for your NMS.

Table 17-1 SNMP MIB and Trap Support

MIB or Trap Support	Description
SNMP core traps	The FWSM sends the following core SNMP traps: • authentication—An SNMP request fails because the NMS did not authenticate with the correct community string. • linkup—A VLAN interface is up. • linkdown—A VLAN interface is down, for example, if you removed the **nameif** command, or the VLAN was removed from the switch configuration. • coldstart—The FWSM is running after a reload.
MIB-II	The FWSM supports browsing of the following groups and tables: • system • interfaces • ip.ipAddrTable
Cisco Firewall MIB	The FWSM supports browsing of the following groups: • cfwEvents • cfwSystem The information in cfwSystem.cfwStatus, which relates to failover status, pertains to the entire device and not just a single context. The FWSM supports the following trap: • cfwSecurityNotification
Cisco Memory Pool MIB	The FWSM supports browsing of the following table: • ciscoMemoryPoolTable—The memory usage described in this table applies only to the FWSM general-purpose processor, and not to the network processors.
Cisco Process MIB	The FWSM supports browsing of the following table: • cpmCPUTotalTable—The CPU usage described in this table applies only to the FWSM general-purpose processor, and not to the network processors.
Cisco Syslog MIB	The FWSM supports the following trap: • clogMessageGenerated You cannot browse this MIB.

Enabling SNMP

The SNMP agent that runs on the FWSM performs two functions:

- Replies to SNMP requests from NMSs.
- Sends traps (event notifications) to NMSs.

To enable the SNMP agent and identify an NMS that can connect to the FWSM, follow these steps:

Step 1 To identify the IP address of the NMS that can connect to the FWSM, enter the following command:

```
FWSM/contexta(config)# snmp-server host interface_name ip_address [trap | poll]
[udp-port port]
```

Specify **trap** or **poll** if you want to limit the NMS to receiving traps only or browsing (polling) only. By default, the NMS can use both functions.

SNMP traps are sent on UDP port 162 by default. You can change the port number using the **udp-port** keyword.

Step 2 To specify the community string, enter the following command:

```
FWSM/contexta(config)# snmp-server community key
```

The SNMP community string is a shared secret between the FWSM and the NMS. The key is a case-sensitive value up to 32 characters in length. Spaces are not permitted. The default is **public**.

Step 3 (Optional) To set the SNMP server location or contact information, enter the following command:

```
FWSM/contexta(config)# snmp-server {contact | location} text
```

Step 4 To enable the FWSM to send traps to the NMS, enter the following command:

```
FWSM/contexta(config)# snmp-server enable traps [all | syslog | firewall | snmp [trap1]
[trap2] [...]]
```

By default, SNMP core traps are enabled (**snmp**). If you do not enter a trap type in the command, **syslog** is the default. To enable or disable all traps, enter the **all** option. For **snmp**, you can identify each trap type separately. See Table 17-1 on page 17-2 for a list of traps.

Step 5 To enable system messages to be sent as traps to the NMS, enter the following command:

```
FWSM/contexta(config)# logging history level
```

You must also enable **syslog** traps using the **snmp-server enable traps** command above.

Step 6 To enable logging, so system messages are generated and can then be sent to an NMS, enter the following command:

```
FWSM/contexta(config)# logging on
```

The following example sets the FWSM to receive requests from host 192.168.3.2 on the inside interface, but the FWSM does not send SNMP traps.

```
FWSM/contexta(config)# snmp-server host 192.168.3.2
FWSM/contexta(config)# snmp-server location building 42
FWSM/contexta(config)# snmp-server contact kim lee
FWSM/contexta(config)# snmp-server community ohwhatakeyisthee
```

Troubleshooting the Firewall Services Module

This section describes how troubleshoot the FWSM, and includes the following topics:

Testing Your Configuration

This section describes how to test connectivity for the single mode FWSM or for each security context. The following steps describe how to ping the FWSM interfaces, and how to allow hosts on one interface to ping through to hosts on another interface.

We recommend that you only enable pinging and debug messages during troubleshooting. When you are done testing the FWSM, follow the steps in the "Disabling the Test Configuration" section on page 17-8.

This section includes:

Enabling ICMP Debug Messages and System Messages

Debug messages and system messages can help you troubleshoot why your pings are not successful. The FWSM only shows ICMP debug messages for pings to the FWSM interfaces, and not for pings through the FWSM to other hosts. To enable debugging and system messages, follow these steps:

Step 1 To show ICMP packet information for pings to the FWSM interfaces, enter the following command:

```
FWSM/contexta(config)# debug icmp trace
```

Step 2 To set system messages to be sent to Telnet or SSH sessions, enter the following command:

```
FWSM/contexta(config)# logging monitor debug
```

You can alternately use **logging buffer debug** to send messages to a buffer, and then view them later using the **show logging** command.

Step 3 To send the system messages to your Telnet or SSH session, enter the following command:

```
FWSM/contexta(config)# terminal monitor
```

Step 4 To enable system messages, enter the following command:

```
FWSM/contexta(config)# logging on
```

The following example shows a successful ping from an external host (209.165.201.2) to the FWSM outside interface (209.165.201.1):

```
FWSM/contexta(config)# debug icmp trace
Inbound ICMP echo reply (len 32 id 1 seq 256) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 512) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 512) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 768) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 768) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 1024) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 1024) 209.165.201.1 > 209.165.201.2
```

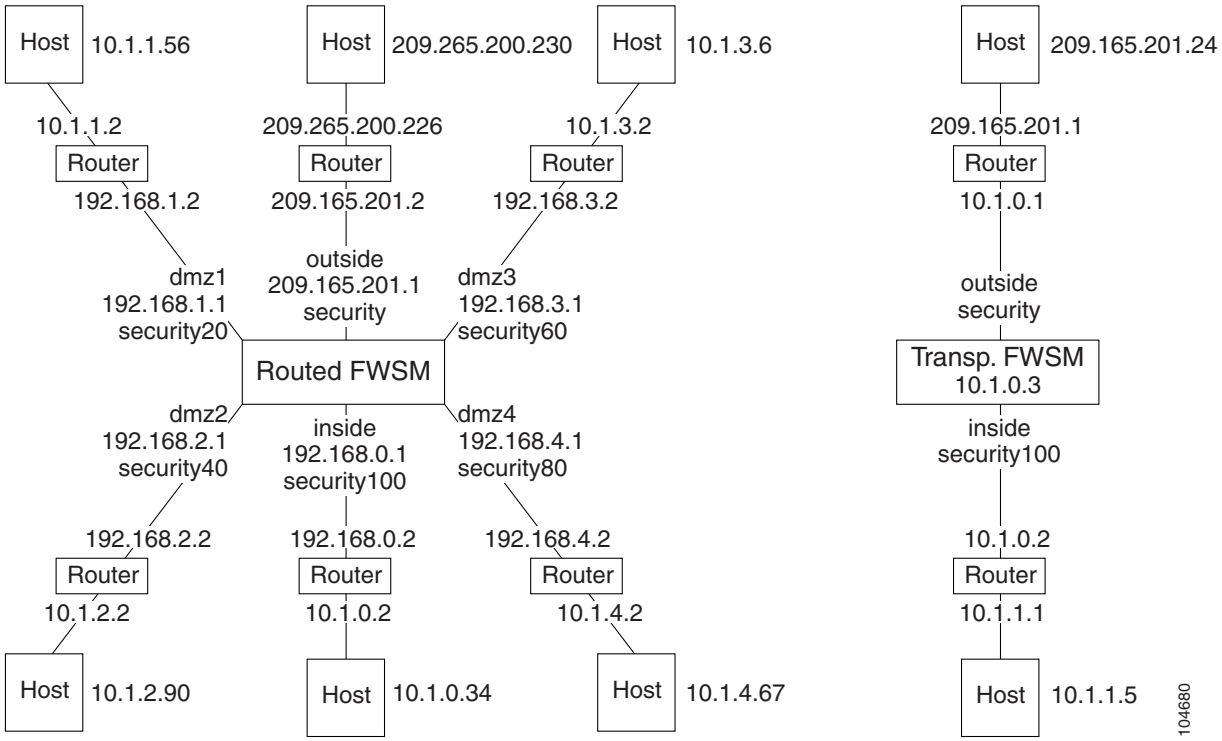
The above example shows the ICMP packet length (32 bytes), the ICMP packet identifier (1), and the ICMP sequence number (the ICMP sequence number starts at 0 and is incremented each time a request is sent).

Pinging FWSM Interfaces

To test that the FWSM interfaces are up and running and that the FWSM and connected routers are routing correctly, you can ping the FWSM interfaces. To ping the FWSM interfaces, follow these steps:

Step 1 Create a sketch of your single mode FWSM or security context showing the interface names, security levels, and IP addresses. The sketch should also include any directly connected routers, and a host on the other side of the router from which you will ping the FWSM. You will use this information for this procedure as well as the procedure in the "Pinging Through the FWSM" section on page 17-7. For example:

Figure 17-1 Network Sketch with Interfaces, Routers, and Hosts



Step 2 To enable ICMP to the FWSM, enter the following command:

```
FWSM/contexta(config)# icmp permit any interface_name
```

Enter this command for each interface you want to test.

Step 3 Ping each FWSM interface from the *directly connected* routers. For transparent mode, ping the management IP address.

This test ensures that the FWSM interfaces are active and that the VLAN configuration is correct.

A ping might fail if the FWSM interface is not active, the VLAN configuration is incorrect, or if a switch between the FWSM and router is down (see Figure 17-2). In this case, no debug messages or system messages appear on the FWSM, because the packet never reaches it.

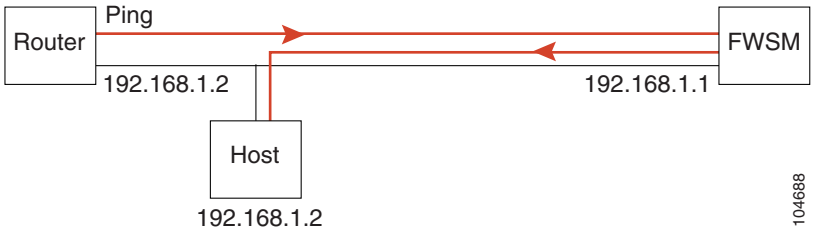
Figure 17-2 Ping Failure at FWSM Interface

If the ping reaches the FWSM, and the FWSM responds, you see debug messages like the following:

```
ICMP echo reply (len 32 id 1 seq 256) 209.165.201.1 > 209.165.201.2
ICMP echo request (len 32 id 1 seq 512) 209.165.201.2 > 209.165.201.1
```

If the ping reply does not return to the router, then you might have a switch loop or redundant IP addresses (see Figure 17-3).

Figure 17-3 Ping Failure Because of IP Addressing Problems

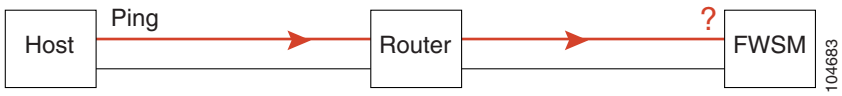


Step 4 Ping each FWSM interface from a remote host. For transparent mode, ping the management IP address.

This test checks that the directly connected router can route the packet between the host and the FWSM, and that the FWSM can correctly route the packet back to the host.

A ping might fail if the FWSM does not have a route back to the host through the intermediate router (see Figure 17-4). In this case, the debug messages show that the ping was successful, but you see system message 110001 indicating a routing failure.

Figure 17-4 Ping Failure Because the FWSM has no Route



Pinging Through the FWSM

After you successfully ping the FWSM interfaces, you should make sure traffic can pass successfully through the FWSM. For routed mode, this test shows that NAT is working correctly. For transparent mode, which does not use NAT, this test confirms that the FWSM is operating correctly; if the ping fails in transparent mode, contact technical support.

You should originate pings from hosts that are normally allowed to access remote networks; you do not need to ping from outside to inside, for example, if you do not allow any outside hosts to access the inside.

To ping between hosts on different interfaces, follow these steps:

Step 1 To add an ACL allowing ICMP from any source host, enter the following command:

```
FWSM/contexta(config)# access-list ICMPTEST extended permit icmp any any
```

Step 2 To assign the ACL to each source interface, enter the following command:

```
FWSM/contexta(config)# access-group ICMPTEST in interface interface_name
```

Repeat this command for each source interface.

Step 3 To enable the ICMP inspection engine, so ICMP responses are allowed back to the source host, enter the following command:

```
FWSM/contexta(config)# fixup protocol icmp
```

Alternatively, you can also apply the ICMPTEST ACL to the destination interface to allow ICMP traffic back through the FWSM.

Step 4 Ping from the host or router through the source interface to another host or router on another interface.

Repeat this step for as many interface pairs as you want to check.

If the ping succeeds, you see a system message confirming the address translation for routed mode (305009 or 305011) and that an ICMP connection was established (302020). You can also enter the **show xlate** and **show conns** commands to view this information.

If the ping fails for transparent mode, contact technical support.

For routed mode, the ping might fail because NAT is not configured correctly (see Figure 17-5). In this case, you see a system message showing that the NAT translation failed (305005 or 305006). If the ping is from an outside host to an inside host, and you do not have a static translation, you see message 106010: deny inbound icmp.

Note The FWSM only shows ICMP debug messages for pings to the FWSM interfaces, and not for pings through the FWSM to other hosts.

Figure 17-5 Ping Failure Because the FWSM is not Translating Addresses

Disabling the Test Configuration

After you complete your testing, disable the test configuration that allows ICMP to and through the FWSM and that prints debug messages. If you leave this configuration in place, it can pose a serious security risk. Debug messages also slow the FWSM performance.

To disable the test configuration, follow these steps:

Step 1 To disable ICMP debug messages, enter the following command:

```
FWSM/contexta(config)# no debug icmp trace
```

Step 2 To disable logging, if desired, enter the following command:

```
FWSM/contexta(config)# no logging on
```

Step 3 To disable ICMP to the FWSM for all interfaces, enter the following command:

```
FWSM/contexta(config)# clear icmp
```

If you want to disable ICMP for a certain interface, use the **no icmp permit** *interface_name* command.

Step 4 To remove the ICMPTEST ACL, and also delete the related **access-group** commands, enter the following command:

```
FWSM/contexta(config)# no access-list ICMPTEST
```

Step 5 (Optional) To disable the ICMP inspection engine, enter the following command:

```
FWSM/contexta(config)# no fixup protocol icmp
```

Reloading the Firewall Services Module

If you need to reload the FWSM, see the following sections:

- Reloading the FWSM from the FWSM CLI, page 17-8
- Reloading the FWSM from the Switch, page 17-9

Reloading the FWSM from the FWSM CLI

In multiple mode, you can only reload from the system execution space. To reload the FWSM from the FWSM CLI, enter the following command:

```
FWSM# reload
```

Reloading the FWSM from the Switch

If you need to reload the FWSM from the switch into the current partition, enter the command for your operating system. See the "Resetting the FWSM or Booting from a Specific Partition" section on page 2-13 for other options.

- For Cisco IOS software, enter the following command:

  ```
  Router# hw-module module mod_num reset
  ```

- For Catalyst OS, enter the following command:

  ```
  Console> (enable) reset mod_num
  ```

Troubleshooting Passwords and AAA

If you forget passwords, or you create a lockout situation because of AAA settings, the following sections describe how to recover:

- Clearing the Application Partition Passwords and AAA Settings, page 17-9
- Recovering the Maintenance Partition Passwords, page 17-10

Clearing the Application Partition Passwords and AAA Settings

If you forget the login and enable passwords, or you create a lockout situation because of AAA settings, you can reset the passwords and portions of AAA configuration to the default values. You must log into the maintenance partition to perform this procedure:

Step 1 To boot the FWSM into the maintenance partition, enter the command for your operating system:

- For Cisco IOS software, enter the following command:

  ```
  Router# hw-module module mod_num reset cf:1
  ```

- For Catalyst OS, enter the following command:

  ```
  Console> (enable) reset mod_num cf:1
  ```

Step 2 To session into the FWSM, enter the command for your operating system:

- For Cisco IOS software, enter the following command:

  ```
  Router# session slot mod_num processor 1
  ```

- For Catalyst OS, enter the following command:

  ```
  Console> (enable) session mod_num
  ```

Step 3 To log into the maintenance partition as root, enter the following command:

```
Login: root
```

Step 4 Enter the password at the prompt:

```
Password: password
```

By default, the password is "cisco."

Step 5 To clear the login and enable passwords, as well as the **aaa authentication console** and **aaa authorization command** commands, enter the following command:

```
root@localhost# clear passwd cf:{4 | 5}
```

Step 6 Follow the screen prompts, as follows:

```
Do you wish to erase the passwords? [yn] y
The following lines will be removed from the configuration:
        enable password 8Ry2YjIyt7RRXU24 encrypted
        passwd 2KFQnbNIdI.2KYOU encrypted
Do you want to remove the commands listed above from the configuration?
[yn] y
Passwords and aaa commands have been erased.
```

Recovering the Maintenance Partition Passwords

If you forget the passwords for the maintenance partition, you can reset them to the default values. You must be logged into the application partition. In multiple mode, you can only reset the passwords from the system execution space.

To reset the maintenance passwords, enter the following command:

```
FWSM# clear mp-passwd
```

Other Troubleshooting Tools

The FWSM provides other troubleshooting tools to be used in conjunction with technical support:

- Viewing Debug Messages, page 17-10
- Capturing Packets, page 17-10
- Viewing the Crash Dump, page 17-11

Viewing Debug Messages

Debug messages can slow the FWSM performance considerably. However, if you are troubleshooting the FWSM, debug messages can be useful. We recommend contacting technical support to help you debug your FWSM. To enable debug messages, see the **debug** commands in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

Capturing Packets

Capturing packets is sometimes useful when troubleshooting connectivity problems or monitoring suspicious activity. The FWSM can track packet information for traffic that passes through the general-purpose processor, including management traffic and inspection engines. The FWSM cannot capture traffic that goes through the network processors (such as most through traffic). We recommend contacting technical support if you want to use the packet capture feature. See the **capture** command in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

Viewing the Crash Dump

If the FWSM crashes, you can view the crash dump information. We recommend contacting technical support if you want to interpret the crash dump. See the **show crashdump** command in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

Common Problems

This section describes common problems with the FWSM, and how you might resolve them.

Symptom When you reset the FWSM from the switch CLI, the system always boots into the maintenance partition.

Possible Cause The default boot partition is set to cf:1.

Recommended Action Change the default boot partition according to the "Setting the Default Boot Partition" section on page 2-13.

Symptom You are unable to log into the maintenance partition with the same password as the application partition.

Possible Cause The application partition and the maintenance partition have different password databases.

Recommended Action Use the password appropriate for your partition. See the "Changing the Passwords" section on page 6-1 for more information.

Symptom Traffic does not pass through the FWSM.

Possible Cause The VLANs are not configured on the switch or are not assigned to the FWSM.

Recommended Action Configure the VLANs and assign them to the FWSM according to the "Assigning VLANs to the Firewall Services Module" section on page 2-2.

Symptom You cannot configure a VLAN interface within a context.

Possible Cause You did not assign that VLAN to the context.

Recommended Action Assign VLANs to contexts according to the "Configuring a Security Context" section on page 5-17.

Symptom You cannot add more than one switched virtual interface (SVI) to the MSFC.

Possible Cause You did not enable multiple SVIs.

Recommended Action Enable multiple SVIs according to the "Adding Switched Virtual Interfaces to the MSFC" section on page 2-5.

Symptom The context configuration was not saved, and was lost when you reloaded.

Possible Cause You did not save each context within the context execution space. If you are configuring contexts at the command line, you did not save the context before you changed to the next context.

Recommended Action Save each context within the context execution space using the **copy run start** command. You cannot save contexts from the system execution space.

Symptom You cannot make a Telnet connection or SSH to the FWSM interface.

Possible Cause You did not enable Telnet or SSH to the FWSM.

Recommended Action Enable Telnet or SSH to the FWSM according to the "Allowing Telnet" section on page 11-1 or the "Allowing SSH" section on page 11-2.

Symptom You cannot ping the FWSM interface.

Possible Cause You did not enable ICMP to the FWSM.

Recommended Action Enable ICMP to the FWSM according to the "Allowing ICMP to and from the FWSM" section on page 11-10.

Symptom You cannot ping through the FWSM, even though the ACL allows it.

Possible Cause You did not enable the ICMP inspection engine or apply ACLs on both the source and destination interfaces.

Recommended Action Because ICMP is a connectionless protocol, the FWSM does not automatically allow returning traffic through. In addition to an ACL on the source interface, you either need to apply an ACL to destination interface to allow replying traffic, or enable the ICMP inspection engine, which treats ICMP connections as stateful connections.

Symptom Traffic does not go through the FWSM from a higher security interface to a lower security interface.

Possible Cause You did not apply an ACL to the higher security interface to allow traffic through. Unlike the PIX firewall, the FWSM does not automatically allow traffic to pass between interfaces.

Recommended Action Apply an ACL to the source interface to allow traffic through. See the "Adding an Extended Access Control List" section on page 10-13.

Symptom Traffic does not pass between two interfaces on the same security level.

Possible Cause You did not enable the feature that allows traffic to pass between interfaces on the same security level.

Recommended Action Enable this feature according to the "Allowing Communication Between Interfaces on the Same Security Level" section on page 6-8.

Symptom When the FWSM fails over, the secondary unit does not pass traffic.

Possible Cause You did not assign the same VLANs for both units.

Recommended Action Make sure to assign the same VLANs to both units in the switch configuration.

Specifications

This chapter lists the specifications of the Firewall Services Module (FWSM) and includes the following sections:

Physical Attributes

Table A-1 lists the physical attributes of the FWSM.

Table A-1 Physical Attributes

Specification	Description
Bandwidth	CEF256 line card with a 6-Gbps path to the Switch Fabric Module (if present) or the 32-Gbps shared bus. With 64-byte Ethernet frames, the FWSM supports 2.84 Mpps throughput; with 1500-byte frames, the FWSM supports 5.456 Gbps throughput.
Memory	• 1 GB RAM. • 128-MB Flash memory.
Modules per switch	Maximum four modules per switch. If you are using failover, you can still only have four modules per switch even if two of them are in standby mode.

Feature Limits

Table A-2 lists the feature limits for the FWSM.

Table A-2 Feature Limits

	Context Mode	
Specification	**Single**	**Multiple**
AAA servers (RADIUS and TACACS+)	16	4 per context
Failover interface monitoring	250	250 divided between all contexts
Filtering servers (Websense Enterprise and Sentian by N2H2)	16	4 per context
Jumbo Ethernet packets	8500 Bytes	8500 Bytes
Security contexts	N/A	100 security contexts (depending on your software license).
Syslog servers	16	4 per context
VLAN interfaces		
Routed Mode	256	256 per context The FWSM has an overall limit of 1000 VLAN interfaces divided between all contexts. You can share outside interfaces between contexts, and in some circumstances, you can share inside interfaces.
Transparent Mode	2	2 per context

Managed System Resources

Table A-3 lists the managed system resources of the FWSM. You can manage these resources per context using the resource manager. See the "Configuring Resource Management" section on page 5-11.

Table A-3 Managed System Resources

	Context Mode	
Specification	**Single**	**Multiple**
MAC addresses (transparent firewall mode only)	64 K	64 K divided between all contexts
Hosts allowed to connect through the FWSM, concurrent	256 K	256 K divided between all contexts
Inspection engine connections, rate	10,000 per second	10,000 per second divided between all contexts
IPSec management connections, concurrent	5	5 per context Maximum of 10 divided between all contexts
NAT translations, concurrent	256 K	256 K divided between all contexts
SSH[1] management connections, concurrent	5	5 per context Maximum of 100 divided between all contexts
System messages, rate	30,000 per second for messages sent to the FWSM terminal or buffer 25,000 per second for messages sent to a syslog server	30,000 per second divided between all contexts for messages sent to the FWSM terminal or buffer 25,000 per second divided between all contexts for messages sent to a syslog server
TCP[2] or UDP[3] connections between any two hosts, including connections between one host and multiple other hosts, concurrent and rate[4]	999,900 100,000 per second	999,900 divided between all contexts 100,000 per second divided between all contexts
Telnet management connections, concurrent	5	5 per context Maximum of 100 connections divided between all contexts.

1. Secure Shell
2. Transmission Control Protocol
3. User Datagram Protocol
4. Because Port Address Translation (PAT) requires a separate translation for each connection, the effective limit of connections using PAT is the translation limit (256K), not the higher connection limit. To use the connection limit, you need to use NAT, which allows multiple connections using the same translation session.

Fixed System Resources

Table A-4 lists the fixed system resources of the FWSM.

Table A-4 Fixed System Resources

Specification	Context Mode	
	Single	**Multiple**
AAA[1] connections, rate	80 per second	80 per second divided between all contexts
ACL logging flows, concurrent	32 K	32 K divided between all contexts
Alias statements	1 K	1 K divided between all contexts
ARP[2] table entries, concurrent	64 K	64 K divided between all contexts
DNS inspections, rate	5000 per second	5000 per second divided between all contexts
Global statements	1,051	1,051 divided between all contexts
HTTP(S) connections, concurrent (for PDM)[3]	16	5 per context Maximum of 16 divided between all contexts
Inspection engine (fixup) statements	32	32 per context[4]
NAT statements	2 K	2 K divided between all contexts
Packet reassembly, concurrent	30,000	30,000 fragments divided between all contexts
Route table entries, concurrent	32 K	32 K divided between all contexts
Shun statements	5 K	5 K divided between all contexts
SIP connections, concurrent	5 K	5 K divided between all contexts
Static NAT statements	2 K	2 K divided between all contexts
TFTP sessions, concurrent[5]	999,100	999,100 divided between all contexts
User authentication sessions, concurrent	50 K	50 K divided between all contexts
User authorization sessions, concurrent	150 K Maximum 15 sessions per user.	150 K divided between all contexts Maximum 15 sessions per user.

1. authentication, authorization, and accounting
2. Address Resolution Protocol
3. PDM uses two HTTPS connections: one for monitoring that is always present, and one for making configuration changes that is used only when you make changes. If all users are making configuration changes at the same time, then the effective number of PDM users is half the available HTTPS connections.
4. This limit includes the following inspection engines that are enabled by default, making the total number of configurable inspection engines 27: TFTP, Sun RPC over UDP, NetBIOS NameServer, XDMCP, and CUSeeMe. The OraServ and RealAudio inspection engines, which are also enabled by default, do not affect this limit.
5. In FWSM Version 1.1, the number of TFTP sessions was limited to 1024 sessions.

Rule Limits

The FWSM supports approximately 80K rules for the entire system in single mode, and 142K rules for multiple mode.

In multiple context mode, each context supports at most 12,130 rules, but the actual number of rules supported in a context might be less, depending on how many contexts you have. A context belongs to one of 12 pools that offers a maximum of 12,130 rules. The FWSM assigns contexts to the pools in the order they are loaded at startup. For example, if you have 12 contexts, each context is assigned to its own pool, and can use 12,130 rules. If you add one more context, then context number 1 and the new context number 13 are both assigned to pool 1, and can use 12,130 rules divided between them; the other 11 contexts continue to use 12,130 rules each. If you delete contexts, the pool membership does not shift, so you might have some unequal distribution until you reboot, at which time the contexts are evenly distributed.

See the "Maximum Number of ACEs" section on page 10-7 for information about memory usage by ACLs.

Note Rules are used up on a first come, first served basis, so one context might use more rules than another context.

Table A-5 lists the maximum number of each rule type.

Table A-5 Rule Limits

	Context Mode	
Specification	**Single**	**Multiple (Maximum per Pool)**
AAA Rules	3,942	606[1]
ACEs[2]	63,078	9,704
Downloaded ACEs for network access authorization	3 K	3 K
Established Rules	788	121
Filter Rules	3,942	606
ICMP[3], Telnet, SSH, and HTTP[4] Rules	2,365	363
Policy NAT ACEs	3,942	606

1. For example, if you have 96 contexts evenly distributed among the 12 pools, so there are 8 contexts per pool, each context can use 75 filter rules, if evenly divided.
2. access control entries
3. Internet Control Message Protocol
4. HyperText Transfer Protocol

Sample Configurations

This chapter illustrates and describes a number of common ways to implement the Firewall Services Module (FWSM). It includes the following topics:

Routed Mode Examples

This section includes the following topics:

Example 1: Security Contexts With Outside Access

This configuration creates three security contexts plus the admin context, each with an inside and an outside interface. The Customer C context includes a DMZ interface where a Websense server for HTTP filtering resides on the service provider premises (see Figure B-1).

Inside hosts can access the Internet through the outside using dynamic NAT or PAT, but no outside hosts can access the inside.

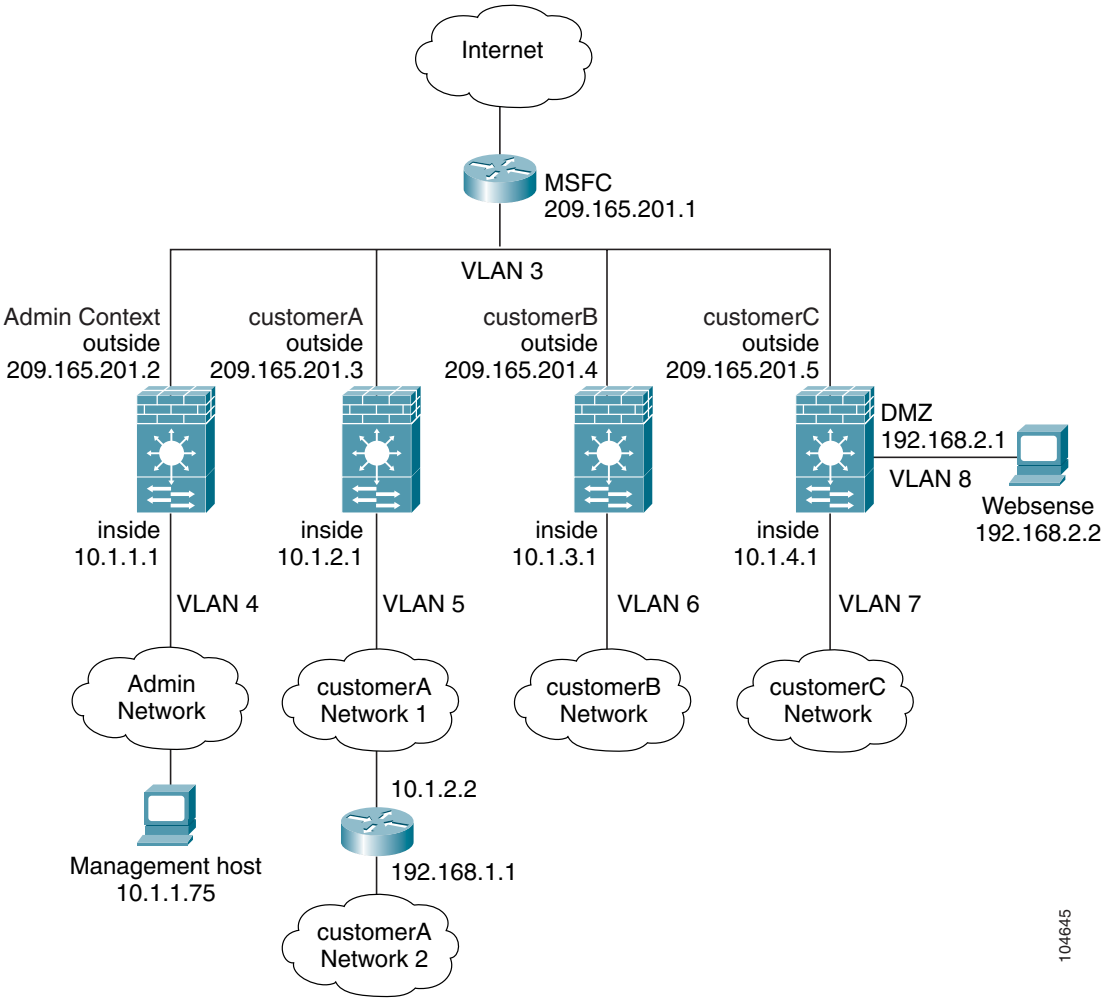
The Customer A context has a second network behind an inside router.

The admin context allows SSH sessions to the FWSM from one host.

Each customer context belongs to a class that limits its resources (gold, silver, or bronze).

Although inside IP addresses can be the same across contexts when the VLANs are unique, keeping them unique is easier to manage.

Figure B-1 Example 1



See the following sections for the configurations for this scenario:

- Example 1: System Configuration, page B-2
- Example 1: Admin Context Configuration, page B-3
- Example 1: Customer A Context Configuration, page B-4
- Example 1: Customer B Context Configuration, page B-4
- Example 1: Customer C Context Configuration, page B-4
- Example 1: Switch Configuration, page B-5

Example 1: System Configuration

You must first enable multiple context mode using the **mode multiple** command. Then enter the activation key to allow more than two contexts using the **activation-key** command. The mode and the activation key are not stored in the configuration file, even though they do endure reboots. If you view the configuration on the FWSM using the **write terminal**, **show startup**, or **show running** commands, the mode displays after the FWSM Version (blank means single mode, "<system>" means you are in multiple mode in the system configuration, and <context> means you are in multiple mode in a context).

```
hostname Farscape
password passw0rd
enable password chr1cht0n
admin-context admin
context admin
    allocate-interface vlan3
    allocate-interface vlan4
    config-url disk://admin.cfg
    class default
context customerA
    description This is the context for customer A
    allocate-interface vlan3
    allocate-interface vlan5
    config-url disk://contexta.cfg
    class gold
context customerB
    description This is the context for customer B
    allocate-interface vlan3
    allocate-interface vlan6
    config-url disk://contextb.cfg
    class silver
context customerC
    description This is the context for customer C
    allocate-interface vlan3
    allocate-interface vlan7-vlan8
    config-url disk://contextc.cfg
    class bronze
class gold
    limit-resource all 7%
    limit-resource rate conns 2000
    limit-resource conns 20000
class silver
    limit-resource all 5%
    limit-resource rate conns 1000
    limit-resource conns 10000
class bronze
    limit-resource all 3%
    limit-resource rate conns 500
    limit-resource conns 5000
```

Example 1: Admin Context Configuration

The host at 10.1.1.75 can access the context using SSH, which requires a certificate to be generated using the **ca generate rsa key** *modulus* command and saved using the **ca save all** command. The certificate is saved in Flash memory.

```
hostname Admin
domain isp
nameif vlan3 outside security0
nameif vlan4 inside security100
passwd secret1969
enable password h1andl0
ip address outside 209.165.201.2 255.255.255.224
ip address inside 10.1.1.1 255.255.255.0
route outside 0 0 209.165.201.1 1
ssh 10.1.1.75 255.255.255.255 inside
nat (inside) 1 10.1.1.0 255.255.255.0
global (outside) 1 209.165.201.10-209.165.201.29 [This context uses dynamic NAT for inside
users that access the outside]
```

```
static (inside,outside) 209.165.201.30 10.1.1.75 netmask 255.255.255.255 [The host at
10.1.1.75 has access to the Websense server in Customer C, so it needs a static
translation for use in Customer C's ACL]
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside [Allows all inside hosts to access the outside
for any IP traffic]
```

Example 1: Customer A Context Configuration

```
nameif vlan3 outside security0
nameif vlan5 inside security100
passwd hell0!
enable password enter55
ip address outside 209.165.201.3 255.255.255.224
ip address inside 10.1.2.1 255.255.255.0
route outside 0 0 209.165.201.1 1
route inside 192.168.1.0 255.255.255.0 10.1.2.2 1 [The Customer A context has a second
network behind an inside router that requires a static route. All other traffic is handled
by the default route pointing to the MSFC.]
nat (inside) 1 10.1.2.0 255.255.255.0
global (outside) 1 interface [This context uses dynamic PAT for inside users that access
that outside. The outside interface address is used for the PAT address]
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside [Allows all inside hosts to access the outside
for any IP traffic]
```

Example 1: Customer B Context Configuration

```
nameif vlan3 outside security0
nameif vlan6 inside security100
passwd tenac10us
enable password defen$e
ip address outside 209.165.201.4 255.255.255.224
ip address inside 10.1.3.1 255.255.255.0
route outside 0 0 209.165.201.1 1
nat (inside) 1 10.1.3.0 255.255.255.0
global (outside) 1 209.165.201.9 netmask 255.255.255.255 [This context uses dynamic PAT
for inside users that access the outside]
access-list INTERNET extended permit tcp any any eq http
access-list INTERNET extended permit tcp any any eq https
access-group INTERNET in interface inside [Inside users can only access HTTP and HTTPS
servers on the outside]
```

Example 1: Customer C Context Configuration

```
nameif vlan3 outside security0
nameif vlan7 inside security100
nameif vlan8 dmz security50
passwd fl0wer
enable password treeh0u$e
ip address outside 209.165.201.5 255.255.255.224
ip address inside 10.1.4.1 255.255.255.0
ip address dmz 192.168.2.1 255.255.255.0
route outside 0 0 209.165.201.1 1
url-server (dmz) vendor websense host 192.168.2.2 url-block block 50
url-cache dst 128
```

```
filter url http 10.1.4.0 255.255.255.0 0 0 [When inside users access an HTTP server, the
FWSM consults with a Websense server to determine if the traffic is allowed]
nat (inside) 1 10.1.4.0 255.255.255.0
global (outside) 1 209.165.201.9 netmask 255.255.255.255 [This context uses dynamic NAT
for inside users that access the outside]
static (dmz,outside) 209.165.201.6 192.168.2.2 netmask 255.255.255.255 [A host on the
admin context requires access to the Websense server for management using pcAnywhere, so
the Websense server requires a static translation]
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside [Allows all inside hosts to access the outside
for any IP traffic. Because there is no NAT from inside to dmz, you do not have to deny
traffic from accessing the dmz.]
access-list MANAGE extended permit tcp host 209.165.201.30 host 209.165.201.6 eq
pcanywhere-data
access-list MANAGE extended permit udp host 209.165.201.30 host 209.165.201.6 eq
pcanywhere-status
access-group MANAGE in interface outside [This ACL allows the management host to use
pcAnywhere on the Websense server]
access-list WEBSENSE extended permit tcp host 192.168.2.2 any eq http [The Websense server
needs to access the Websense updater server on the outside]
access-group WEBSENSE in interface dmz
```

Example 1: Switch Configuration

The following lines in the Cisco IOS switch configuration relate to the FWSM:

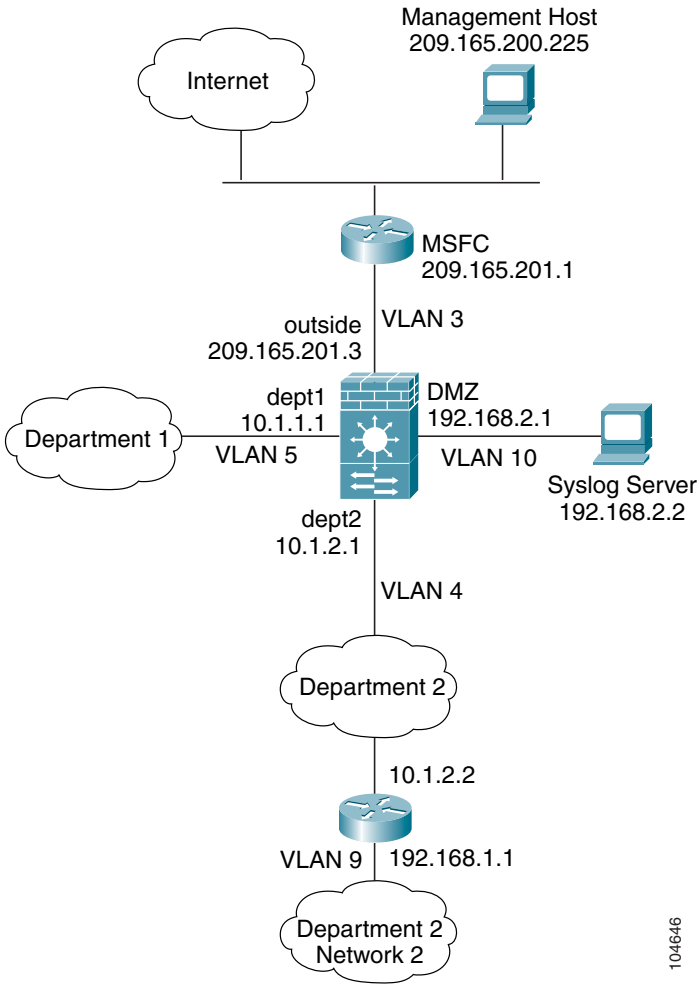
```
...
firewall module 8 vlan-group 1
firewall vlan-group 1 3-8
interface vlan 3
    ip address 209.165.201.1 255.255.255.224
    no shut
...
```

Example 2: Single Mode Using Same Security Level

This configuration creates three internal interfaces. Two of the interfaces connect to departments that are on the same security level, which allows all hosts to communicate without using NAT. The DMZ interface hosts a Syslog server. The management host on the outside needs access to the Syslog server and the FWSM. To connect to the FWSM, the host uses a VPN connection. The FWSM uses RIP on the inside interfaces to learn routes. Because the FWSM does not advertise routes with RIP, the MSFC needs to use static routes for FWSM traffic (see Figure B-2).

The Department networks are allowed to access the Internet, and use PAT.

Figure B-2 Example 2



See the following sections for the configurations for this scenario:

- Example 2: FWSM Configuration, page B-6
- Example 2: Switch Configuration, page B-7

Example 2: FWSM Configuration

```
nameif vlan3 outside security0
nameif vlan4 dept2 security100
nameif vlan5 dept1 security100
nameif vlan10 dmz security50
passwd g00fba11
enable password gen1u$
hostname Buster
same-security-traffic permit inter-interface
ip address outside 209.165.201.3 255.255.255.224
ip address dept2 10.1.2.1 255.255.255.0
ip address dept1 10.1.1.1 255.255.255.0
ip address dmz 192.168.2.1 255.255.255.0
route outside 0 0 209.165.201.1 1
nat (dept1) 1 10.1.1.0 255.255.255.0
nat (dept2) 1 10.1.2.0 255.255.255.0
```

```
global (outside) 1 209.165.201.9 netmask 255.255.255.255 [The dept1 and dept2 networks use
PAT when accessing the outside]
static (dmz,outside) 209.165.201.5 192.168.2.2 netmask 255.255.255.255 [The syslog server
needs a static translation so the outside management host can access the server]
access-list DEPTS extended permit ip any any
access-group DEPTS in interface dept1
access-group DEPTS in interface dept2 [Allows all dept1 and dept2 hosts to access the
outside for any IP traffic]
access-list MANAGE extended permit tcp host 209.165.200.225 host 209.165.201.5 eq telnet
access-group MANAGE in interface outside [This ACL allows the management host to access
the syslog server]
rip dept2 default version 2 authentication md5 scorpius 1 [Advertises the FWSM IP address
as the default gateway for the downstream router. The FWSM does not advertise a default
route to the MSFC.]
rip dept2 passive version 2 authentication md5 scorpius 1 [Listens for RIP updates from
the downstream router. The FWSM does not listen for RIP updates from the MSFC because a
default route to the MSFC is all that is required.]
isakmp policy 1 authentication pre-share [The client uses a pre-shared key to connect to
the FWSM over IPSec. The key is the password in the username command below.]
isakmp policy 1 encryption 3des
isakmp policy 1 group 2
isakmp policy 1 hash sha
isakmp enable outside
crypto ipsec transform-set vpn_client esp-3des esp-sha-hmac
username admin password passw0rd
crypto ipsec transform-set vpn esp-3des esp-sha-hmac
crypto dynamic-map vpn_client 1 set transform-set vpn
crypto map telnet_tunnel 1 ipsec-isakmp dynamic vpn_client
crypto map telnet_tunnel interface outside
crypto map telnet_tunnel client authentication LOCAL
ip local pool client_pool 10.1.1.2
access-list VPN_SPLIT extended permit ip host 209.165.201.3 host 10.1.1.2
vpngroup admin address-pool client_pool
vpngroup admin split-tunnel VPN_SPLIT
vpngroup admin password $ecure23
telnet 10.1.1.2 255.255.255.255 outside
telnet timeout 30
logging trap 5
logging host dmz 192.168.2.2 [System messages are sent to the syslog server on the DMZ
network]
logging on
```

Example 2: Switch Configuration

The following lines in the switch configuration relate to the FWSM:

Catalyst OS on the supervisor:

```
set vlan 3-5,9,10 firewall-vlan 8
```

Cisco IOS software on the MSFC:

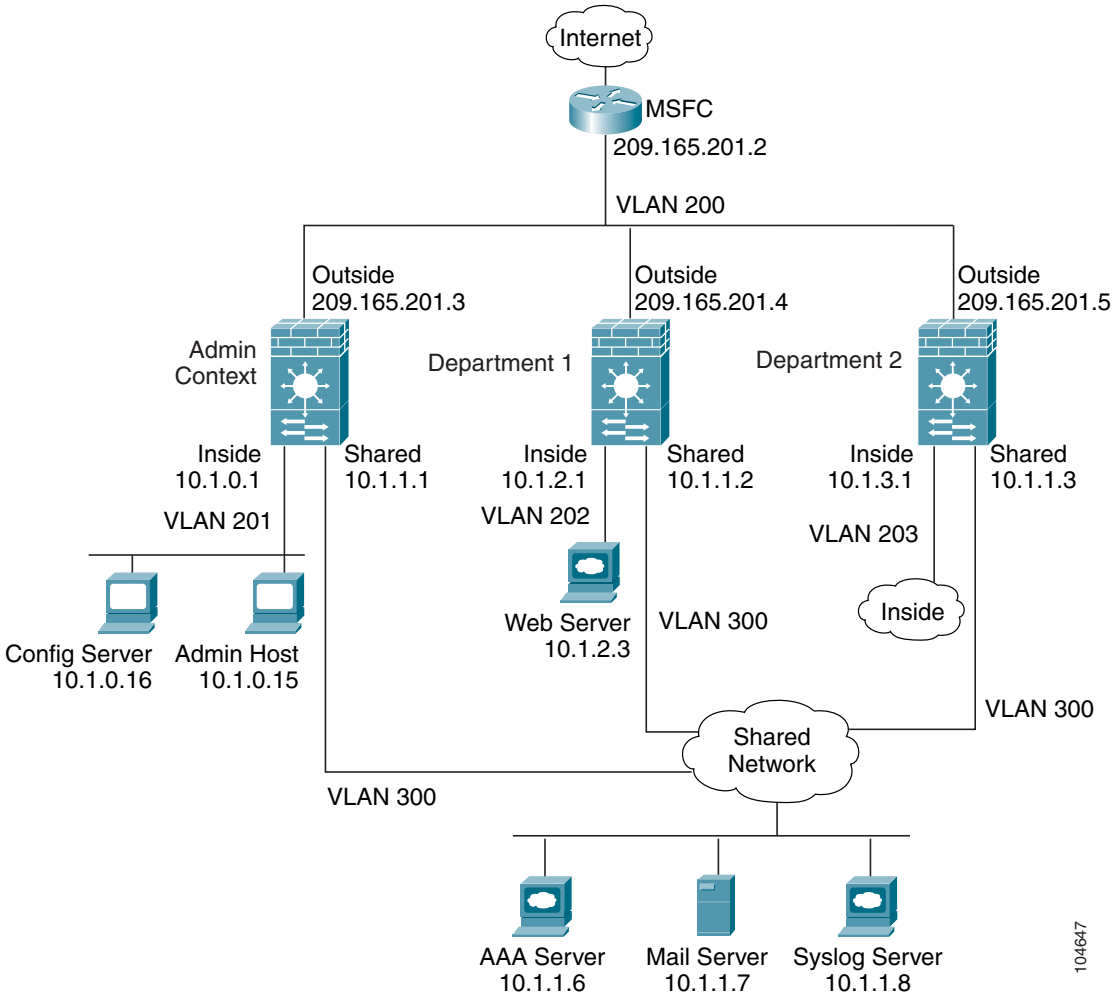
```
interface vlan 3
    ip address 209.165.201.1 255.255.255.224
    no shut
...
```

Example 3: Shared Resources for Multiple Contexts

This configuration includes multiple contexts for multiple departments within a company. Each department has its own security context so that each department can have its own security policy. However, the syslog, mail, and AAA servers are shared across all departments. These servers are placed on a shared VLAN (see Figure B-3).

Department 1 has a web server that outside users who are authenticated by the AAA server can access.

Figure B-3 Example 3



See the following sections for the configurations for this scenario:

- Example 3: System Configuration, page B-9
- Example 3: Admin Context Configuration, page B-9
- Example 3: Department 1 Context Configuration, page B-10
- Example 3: Department 2 Context Configuration, page B-11
- Example 3: Switch Configuration, page B-11

Example 3: System Configuration

You must first enable multiple context mode using the **mode multiple** command. Then enter the activation key to allow more than two contexts using the **activation-key** command. The mode and the activation key are not stored in the configuration file, even though they do endure reboots. If you view the configuration on the FWSM using the **write terminal**, **show startup**, or **show running** commands, the mode displays after the FWSM Version (blank means single mode, "<system>" means you are in multiple mode in the system configuration, and <context> means you are in multiple mode in a context).

```
hostname Ubik
password pkd55
enable password deckard69
admin-context admin
context admin
    allocate-interface vlan200
    allocate-interface vlan201
    allocate-interface vlan300
    config-url disk://admin.cfg
context department1
    allocate-interface vlan200
    allocate-interface vlan202
    allocate-interface vlan300
    config-url ftp://admin:passw0rd@10.1.0.16/dept1.cfg
context department2
    allocate-interface vlan200
    allocate-interface vlan203
    allocate-interface vlan300
    config-url ftp://admin:passw0rd@10.1.0.16/dept2.cfg
```

Example 3: Admin Context Configuration

```
hostname Admin
nameif vlan200 outside security0
nameif vlan201 inside security100
nameif vlan300 shared security50
passwd v00d00
enable password d011
ip address outside 209.165.201.3 255.255.255.224
ip address inside 10.1.0.1 255.255.255.0
ip address shared 10.1.1.1 255.255.255.0
route outside 0 0 209.165.201.2 1
nat (inside) 1 10.1.0.0 255.255.255.0
global (outside) 1 209.165.201.6 netmask 255.255.255.255 [This context uses PAT for inside
users that access the outside]
global (shared) 1 10.1.1.30 [This context uses PAT for inside users that access the shared
network]
static (inside,outside) 209.165.201.7 10.1.0.15 netmask 255.255.255.255 [Because this host
can access the web server in the Department 1 context, it requires a static translation]
static (inside,shared) 10.1.1.78 10.1.0.15 netmask 255.255.255.255 [Because this host has
management access to the servers on the Shared interface, it requires a static translation
to be used in an ACL]
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside [Allows all inside hosts to access the outside
and shared network for any IP traffic]
access-list SHARED extended permit ip host 10.1.1.78 any
access-list SHARED extended permit tcp host 10.1.1.30 host 10.1.1.7 eq smtp
access-group SHARED out interface shared [This ACL allows only mail traffic from the
inside network to exit out the shared interface, but allows the admin host to access any
server. Note that the translated addresses are used.]
```

```
telnet 10.1.0.15 255.255.255.255 inside [Allows 10.1.0.15 to access the admin context
using Telnet. From the admin context, you can access all other contexts.]
aaa-server AAA-SERVER protocol tacacs+
aaa-server AAA-SERVER (shared) host 10.1.1.6 TheUauthKey
aaa authentication telnet console AAA-SERVER [The host at 10.1.0.15 must authenticate with
the AAA server to log in]
logging trap 6
logging host shared 10.1.1.8 [System messages are sent to the syslog server on the Shared
network]
logging on
```

Example 3: Department 1 Context Configuration

```
nameif vlan200 outside security0
nameif vlan202 inside security100
nameif vlan300 shared security50
passwd cugel
enable password rhialto
ip address outside 209.165.201.4 255.255.255.224
ip address inside 10.1.2.1 255.255.255.0
ip address shared 10.1.1.2 255.255.255.0
nat (inside) 1 10.1.2.0 255.255.255.0
global (outside) 1 209.165.201.8 netmask 255.255.255.255 [The inside network uses PAT when
accessing the outside]
global (shared) 1 10.1.1.31-10.1.1.37 [The inside network uses dynamic NAT when accessing
the shared network]
static (inside,outside) 209.165.201.9 10.1.2.3 netmask 255.255.255.255 [The web server can
be accessed from outside and requires a static translation]
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside [Allows all inside hosts to access the outside
and shared network for any IP traffic]
access-list WEBSERVER extended permit ip host 209.165.201.7 host 209.165.201.9 [This ACE
allows the management host (its translated address) on the admin context to access the web
server for management (it can use any IP protocol)]
access-list WEBSERVER extended permit tcp any eq http host 209.165.201.9 eq http [This ACE
allows any outside address to access the web server with HTTP]
access-group WEBSERVER in interface outside
access-list MAIL extended permit tcp host 10.1.1.31 eq smtp host 10.1.1.7 eq smtp
access-list MAIL extended permit tcp host 10.1.1.32 eq smtp host 10.1.1.7 eq smtp
access-list MAIL extended permit tcp host 10.1.1.33 eq smtp host 10.1.1.7 eq smtp
access-list MAIL extended permit tcp host 10.1.1.34 eq smtp host 10.1.1.7 eq smtp
access-list MAIL extended permit tcp host 10.1.1.35 eq smtp host 10.1.1.7 eq smtp
access-list MAIL extended permit tcp host 10.1.1.36 eq smtp host 10.1.1.7 eq smtp
access-list MAIL extended permit tcp host 10.1.1.37 eq smtp host 10.1.1.7 eq smtp
access-group MAIL out interface shared [This ACL allows only mail traffic from the inside
network to exit out the shared interface. Note that the translated addresses are used.]
aaa-server AAA-SERVER protocol tacacs+
aaa-server AAA-SERVER (shared) host 10.1.1.6 TheUauthKey
aaa authentication match WEBSERVER outside AAA-SERVER [All traffic matching the WEBSERVER
ACL must authenticate with the AAA server]
logging trap 4
logging host shared 10.1.1.8 [System messages are sent to the syslog server on the Shared
network]
logging on
```

Example 3: Department 2 Context Configuration

```
nameif vlan200 outside security0
nameif vlan203 inside security100
nameif vlan300 shared security50
passwd maz1r1an
enable password ly0ne$$e
ip address outside 209.165.201.5 255.255.255.224
ip address inside 10.1.3.1 255.255.255.0
ip address shared 10.1.1.3 255.255.255.0
route outside 0 0 209.165.201.2 1
nat (inside) 1 10.1.3.0 255.255.255.0
global (outside) 1 209.165.201.10 netmask 255.255.255.255 [The inside network uses PAT
when accessing the outside]
global (shared) 1 10.1.1.38 [The inside network uses PAT when accessing the shared
network]
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside [Allows all inside hosts to access the outside
and shared network for any IP traffic]
access-list MAIL extended permit tcp host 10.1.1.38 host 10.1.1.7 eq smtp
access-group MAIL out interface shared [This ACL allows only mail traffic from the inside
network to exit out the shared interface. Note that the translated PAT address is used.]
logging trap 3
logging host shared 10.1.1.8 [System messages are sent to the syslog server on the Shared
network]
logging on
```

Example 3: Switch Configuration

The following lines in the Cisco IOS switch configuration relate to the FWSM:

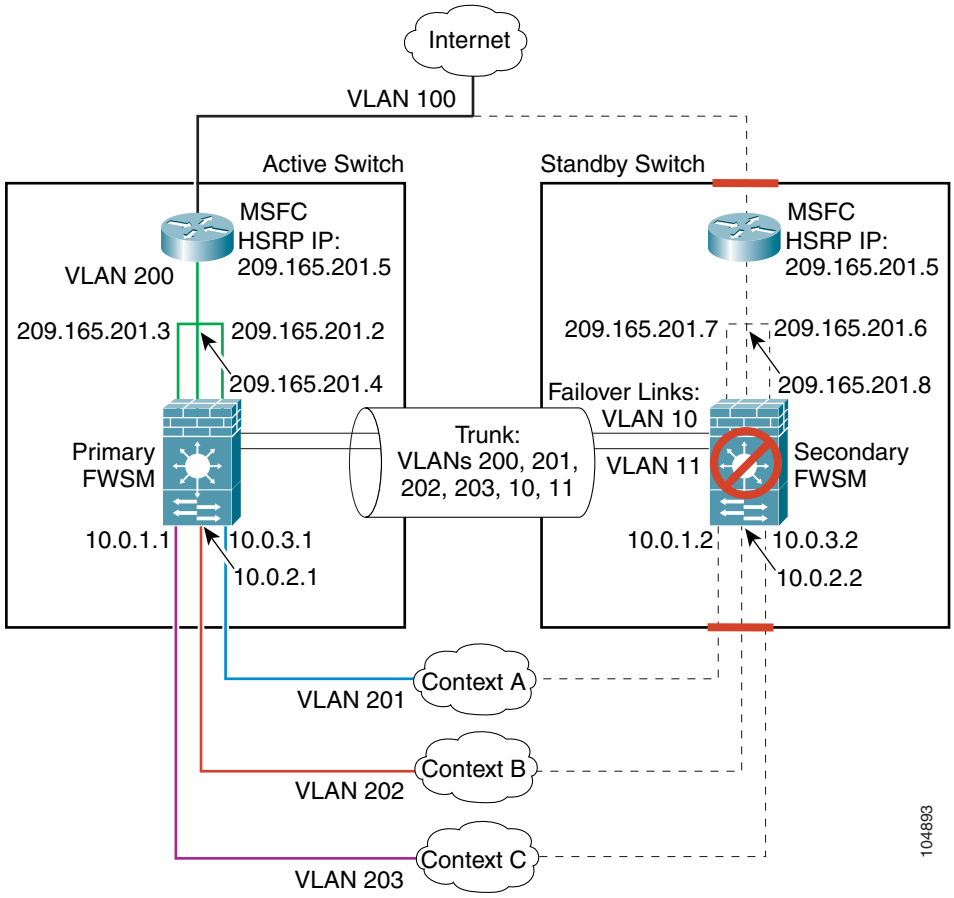
```
...
firewall module 6 vlan-group 1
firewall vlan-group 1 200-203,300
interface vlan 200
    ip address 209.165.201.2 255.255.255.224
    no shut
...
```

Example 4: Failover

This configuration shows a routed, multiple context mode FWSM in one switch, and another FWSM in a second switch acting as a backup (see Figure B-4). Each context (A, B, and C) monitors the inside interface, and context A, which is the admin context, also monitors the outside interface. Because the outside interface is shared among all contexts, monitoring in one context benefits all contexts.

The secondary FWSM is also in routed, multiple context mode, and has the same software version.

Figure B-4 Example 4



See the following sections for the configurations for this scenario:

- Example 4: Primary FWSM Configuration, page B-12
- Example 4: Secondary FWSM System Configuration, page B-14
- Example 4: Switch Configuration, page B-14

Example 4: Primary FWSM Configuration

The following sections include the configuration for the primary FWSM:

- Example 4: System Configuration (Primary), page B-12
- Example 4: Context A Configuration (Primary), page B-13
- Example 4: Context B Configuration (Primary), page B-13
- Example 4: Context C Configuration (Primary), page B-14

Example 4: System Configuration (Primary)

You must first enable multiple context mode using the **mode multiple** command. Then enter the activation key to allow more than two contexts using the **activation-key** command. The mode and the activation key are not stored in the configuration file, even though they do endure reboots. If you view

the configuration on the FWSM using the **write terminal**, **show startup**, or **show running** commands, the mode displays after the FWSM Version (blank means single mode, "<system>" means you are in multiple mode in the system configuration, and <context> means you are in multiple mode in a context).

```
hostname primary
enable password farscape
password crichton
failover lan interface faillink vlan 10
failover link statelink vlan 11
failover lan unit primary
failover interface ip faillink 192.168.253.1 255.255.255.252 standby 192.168.253.2
failover interface ip statelink 192.168.253.5 255.255.255.252 standby 192.168.253.6
failover interface-policy 50%
failover replication http
failover
admin-context contexta
context contexta
    allocate-interface vlan200
    allocate-interface vlan201
    config-url disk://contexta.cfg
context contextb
    allocate-interface vlan200
    allocate-interface vlan202
    config-url ftp://admin:passw0rd@10.0.3.16/contextb.cfg
context contextc
    allocate-interface vlan200
    allocate-interface vlan203
    config-url ftp://admin:passw0rd@10.0.3.16/contextc.cfg
```

Example 4: Context A Configuration (Primary)

```
nameif vlan200 outside security0
nameif vlan201 inside security100
passwd secret1969
enable password h1and10
ip address outside 209.165.201.2 255.255.255.224 standby 209.165.201.6
ip address inside 10.0.3.1 255.255.255.0 standby 10.0.3.2
monitor-interface inside
monitor-interface outside
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
global (outside) 1 209.165.201.10 netmask 255.255.255.224 [This context uses dynamic PAT
for inside users that access the outside]
route outside 0 0 209.165.201.5 1
telnet 10.0.3.75 255.255.255.255 inside
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside [Allows all inside hosts to access the outside
for any IP traffic]
```

Example 4: Context B Configuration (Primary)

```
nameif vlan200 outside security0
nameif vlan202 inside security100
passwd secret1978
enable password 7samura1
ip address outside 209.165.201.4 255.255.255.224 standby 209.165.201.8
ip address inside 10.0.2.1 255.255.255.0 standby 10.0.2.2
monitor-interface inside
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
global (outside) 1 209.165.201.11 netmask 255.255.255.224 [This context uses dynamic PAT
for inside users that access the outside]
route outside 0 0 209.165.201.5 1
```

```
telnet 10.0.2.14 255.255.255.255 inside
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside [Allows all inside hosts to access the outside
for any IP traffic]
```

Example 4: Context C Configuration (Primary)

```
nameif vlan200 outside security0
nameif vlan203 inside security100
passwd secret0997
enable password strayd0g
ip address outside 209.165.201.3 255.255.255.224 standby 209.165.201.7
ip address inside 10.0.1.1 255.255.255.0 standby 10.0.1.2
monitor-interface inside
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
global (outside) 1 209.165.201.12 netmask 255.255.255.224 [This context uses dynamic PAT
for inside users that access the outside]
route outside 0 0 209.165.201.5 1
telnet 10.0.1.65 255.255.255.255 inside
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside [Allows all inside hosts to access the outside
for any IP traffic]
```

Example 4: Secondary FWSM System Configuration

You do not need to configure any contexts, just the following minimal configuration for the system.

You must first enable multiple context mode using the **mode multiple** command. Then enter the activation key to allow more than two contexts using the **activation-key** command. The mode and the activation key are not stored in the configuration file, even though they do endure reboots. If you view the configuration on the FWSM using the **write terminal**, **show startup**, or **show running** commands, the mode displays after the FWSM Version (blank means single mode, "<system>" means you are in multiple mode in the system configuration, and <context> means you are in multiple mode in a context).

```
failover lan interface faillink vlan 10
failover interface ip faillink 192.168.253.1 255.255.255.252 standby 192.168.253.2
failover lan unit secondary
failover
```

Example 4: Switch Configuration

The following lines in the Cisco IOS switch configuration on both switches relate to the FWSM. For information about configuring redundancy for the switch, see the switch documentation.

```
...
firewall module 1 vlan-group 1
firewall vlan-group 1 10,11,200-203
interface vlan 200
    ip address 209.165.201.1 255.255.255.224
    standby 200 ip 209.165.201.5
    standby 200 priority 110
    standby 200 preempt
    standby 200 timers 5 15
    standby 200 authentication Secret
    no shut
interface range gigabitethernet 2/1-3
    channel-group 2 mode on
    switchport trunk encapsulation dot1q
    no shut
...
```

Transparent Mode Examples

This section includes the following topics:

Example 5: Security Contexts With Outside Access

This configuration creates three security contexts plus the admin context. Each context allows OSPF traffic to pass between the inside and outside routers (see Figure B-5).

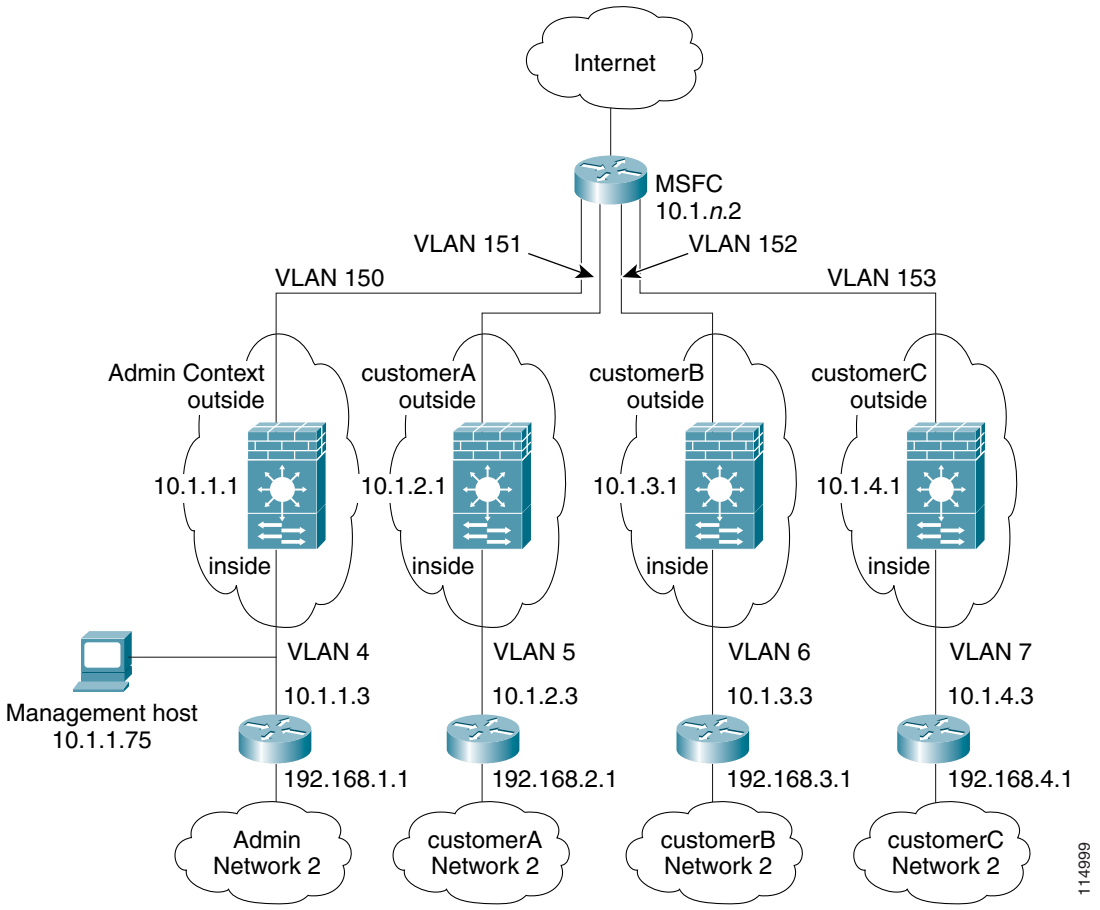
Inside hosts can access the Internet through the outside, but no outside hosts can access the inside.

The admin context allows SSH sessions to the FWSM from one host.

Each customer context belongs to a class that limits its resources (gold, silver, or bronze).

Although inside IP addresses can be the same across contexts, keeping them unique is easier to manage.

Figure B-5 Example 5



See the following sections for the configurations for this scenario:

- Example 1: System Configuration, page B-2
- Example 5: System Configuration, page B-16
- Example 5: Admin Context Configuration, page B-17
- Example 5: Customer A Context Configuration, page B-17
- Example 5: Customer B Context Configuration, page B-17
- Example 5: Customer C Context Configuration, page B-18
- Example 5: Switch Configuration, page B-18

Example 5: System Configuration

You must first enable multiple context mode using the **mode multiple** command. Then enter the activation key to allow more than two contexts using the **activation-key** command. The mode and the activation key are not stored in the configuration file, even though they do endure reboots. If you view the configuration on the FWSM using the **write terminal**, **show startup**, or **show running** commands, the mode displays after the FWSM Version (blank means single mode, "<system>" means you are in multiple mode in the system configuration, and <context> means you are in multiple mode in a context).

```
firewall transparent
hostname Farscape
password passw0rd
enable password chr1cht0n
admin-context admin
context admin
    allocate-interface vlan150
    allocate-interface vlan4
    config-url disk://admin.cfg
    class default
context customerA
    description This is the context for customer A
    allocate-interface vlan151
    allocate-interface vlan5
    config-url disk://contexta.cfg
    class gold
context customerB
    description This is the context for customer B
    allocate-interface vlan152
    allocate-interface vlan6
    config-url disk://contextb.cfg
    class silver
context customerC
    description This is the context for customer C
    allocate-interface vlan153
    allocate-interface vlan7
    config-url disk://contextc.cfg
    class bronze
class gold
    limit-resource all 7%
    limit-resource rate conns 2000
    limit-resource conns 20000
class silver
    limit-resource all 5%
    limit-resource rate conns 1000
    limit-resource conns 10000
class bronze
    limit-resource all 3%
    limit-resource rate conns 500
    limit-resource conns 5000
```

Example 5: Admin Context Configuration

The host at 10.1.1.75 can access the context using SSH, which requires a certificate to be generated using the **ca generate rsa key** *modulus* command and saved using the **ca save all** command. The certificate is saved in Flash memory.

```
hostname Admin
domain isp
nameif vlan150 outside security0
nameif vlan4 inside security100
passwd secret1969
enable password h1and10
ip address 10.1.1.1 255.255.255.0
route outside 0 0 10.1.1.2 1
ssh 10.1.1.75 255.255.255.255 inside
access-list INTERNET extended permit 89 any any
access-list INTERNET extended permit ip any any
access-list OSPF extended permit 89 any any
access-group INTERNET in interface inside [Allows all inside hosts to access the outside
for any IP traffic. Also allows OSPF.]
access-group OSPF in interface outside [Allows OSPF.]
```

Example 5: Customer A Context Configuration

```
nameif vlan151 outside security0
nameif vlan5 inside security100
passwd hell0!
enable password enter55
ip address 10.1.2.1 255.255.255.0
route outside 0 0 10.1.2.2 1
access-list INTERNET extended permit 89 any any
access-list INTERNET extended permit ip any any
access-list OSPF extended permit 89 any any
access-group INTERNET in interface inside [Allows all inside hosts to access the outside
for any IP traffic. Also allows OSPF.]
access-group OSPF in interface outside [Allows OSPF.]
```

Example 5: Customer B Context Configuration

```
nameif vlan152 outside security0
nameif vlan6 inside security100
passwd tenac10us
enable password defen$e
ip address 10.1.3.1 255.255.255.0
route outside 0 0 10.1.3.2 1
access-list INTERNET extended permit 89 any any
access-list INTERNET extended permit ip any any
access-list OSPF extended permit 89 any any
access-group INTERNET in interface inside [Allows all inside hosts to access the outside
for any IP traffic. Also allows OSPF.]
access-group OSPF in interface outside [Allows OSPF.]
```

Example 5: Customer C Context Configuration

```
nameif vlan153 outside security0
nameif vlan7 inside security100
passwd fl0wer
enable password treeh0u$e
ip address 10.1.4.1 255.255.255.0
route outside 0 0 10.1.4.2 1
access-list INTERNET extended permit 89 any any
access-list INTERNET extended permit ip any any
access-list OSPF extended permit 89 any any
access-group INTERNET in interface inside [Allows all inside hosts to access the outside
for any IP traffic. Also allows OSPF.]
access-group OSPF in interface outside [Allows OSPF.]
```

Example 5: Switch Configuration

The following lines in the Cisco IOS switch configuration relate to the FWSM:

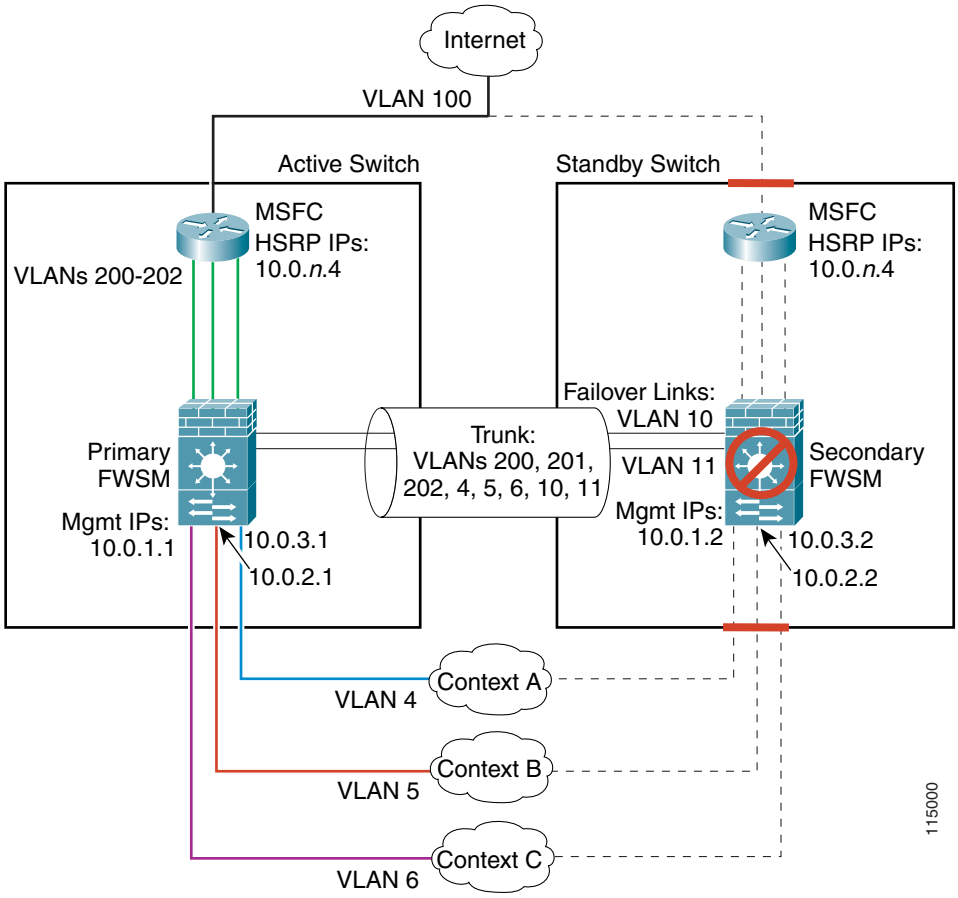
```
...
firewall multiple-vlan-interfaces
firewall module 8 vlan-group 1
firewall vlan-group 1 4-7,150-153
interface vlan 150
    ip address 10.1.1.2 255.255.255.0
    no shut
interface vlan 151
    ip address 10.1.2.2 255.255.255.0
    no shut
interface vlan 152
    ip address 10.1.3.2 255.255.255.0
    no shut
interface vlan 153
    ip address 10.1.4.2 255.255.255.0
    no shut
...
```

Example 6: Failover

This configuration shows a transparent, multiple context mode FWSM in one switch, and another FWSM in a second switch acting as a backup (see Figure B-4). Each context (A, B, and C) monitors the inside interface and outside interface.

The secondary FWSM is also in transparent, multiple context mode, and has the same software version.

Figure B-6 Example 6



See the following sections for the configurations for this scenario:

- Example 6: Primary FWSM Configuration, page B-19
- Example 6: Secondary FWSM System Configuration, page B-21
- Example 6: Switch Configuration, page B-21

Example 6: Primary FWSM Configuration

The following sections include the configuration for the primary FWSM:

- Example 6: System Configuration (Primary), page B-19
- Example 6: Context A Configuration (Primary), page B-20
- Example 6: Context B Configuration (Primary), page B-20
- Example 6: Context C Configuration (Primary), page B-21

Example 6: System Configuration (Primary)

You must first enable multiple context mode using the **mode multiple** command. Then enter the activation key to allow more than two contexts using the **activation-key** command. The mode and the activation key are not stored in the configuration file, even though they do endure reboots. If you view

the configuration on the FWSM using the **write terminal**, **show startup**, or **show running** commands, the mode displays after the FWSM Version (blank means single mode, "<system>" means you are in multiple mode in the system configuration, and <context> means you are in multiple mode in a context).

```
firewall transparent
hostname primary
enable password farscape
password crichton
failover lan interface faillink vlan 10
failover link statelink vlan 11
failover lan unit primary
failover interface ip faillink 192.168.253.1 255.255.255.252 standby 192.168.253.2
failover interface ip statelink 192.168.253.5 255.255.255.252 standby 192.168.253.6
failover interface-policy 1
failover replication http
failover
admin-context contexta
context contexta
    allocate-interface vlan200
    allocate-interface vlan4
    config-url disk://contexta.cfg
context contextb
    allocate-interface vlan201
    allocate-interface vlan5
    config-url ftp://admin:passw0rd@10.0.3.16/contextb.cfg
context contextc
    allocate-interface vlan202
    allocate-interface vlan6
    config-url ftp://admin:passw0rd@10.0.3.16/contextc.cfg
```

Example 6: Context A Configuration (Primary)

```
nameif vlan200 outside security0
nameif vlan4 inside security100
passwd secret1969
enable password h1andl0
ip address 10.0.3.1 255.255.255.0 standby 10.0.3.2
monitor-interface inside
monitor-interface outside
route outside 0 0 10.0.3.4 1
telnet 10.0.3.75 255.255.255.255 inside
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside [Allows all inside hosts to access the outside
for any IP traffic]
access-list BPDU ethertype permit bpdu
access-group BPDU in interface inside
access-group BPDU in interface outside
```

Example 6: Context B Configuration (Primary)

```
nameif vlan201 outside security0
nameif vlan5 inside security100
passwd secret1978
enable password 7samura1
ip address inside 10.0.2.1 255.255.255.0 standby 10.0.2.2
monitor-interface inside
monitor-interface outside
route outside 0 0 10.0.2.4 1
telnet 10.0.2.14 255.255.255.255 inside
access-list INTERNET extended permit ip any any
```

```
access-group INTERNET in interface inside [Allows all inside hosts to access the outside
for any IP traffic]
access-list BPDU ethertype permit bpdu
access-group BPDU in interface inside
access-group BPDU in interface outside
```

Example 6: Context C Configuration (Primary)

```
nameif vlan202 outside security0
nameif vlan6 inside security100
passwd secret0997
enable password strayd0g
ip address inside 10.0.1.1 255.255.255.0 standby 10.0.1.2
monitor-interface inside
monitor-interface outside
route outside 0 0 10.0.1.4 1
telnet 10.0.1.65 255.255.255.255 inside
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside [Allows all inside hosts to access the outside
for any IP traffic]
access-list BPDU ethertype permit bpdu
access-group BPDU in interface inside
access-group BPDU in interface outside
```

Example 6: Secondary FWSM System Configuration

You do not need to configure any contexts, just the following minimal configuration for the system.

You must first enable multiple context mode using the **mode multiple** command. Then enter the activation key to allow more than two contexts using the **activation-key** command. The mode and the activation key are not stored in the configuration file, even though they do endure reboots. If you view the configuration on the FWSM using the **write terminal**, **show startup**, or **show running** commands, the mode displays after the FWSM Version (blank means single mode, "<system>" means you are in multiple mode in the system configuration, and <context> means you are in multiple mode in a context).

```
firewall transparent
failover lan interface faillink vlan 10
failover interface ip faillink 192.168.253.1 255.255.255.252 standby 192.168.253.2
failover lan unit secondary
failover
```

Example 6: Switch Configuration

The following lines in the Cisco IOS switch configuration on both switches relate to the FWSM. For information about configuring redundancy for the switch, see the switch documentation.

```
...
firewall multiple-vlan-interfaces
firewall module 1 vlan-group 1
firewall vlan-group 1 4-6,10,11,200-202
interface vlan 200
    ip address 10.0.1.3 255.255.255.0
    standby 200 ip 10.0.1.4
    standby 200 priority 110
    standby 200 preempt
    standby 200 timers 5 15
    standby 200 authentication Secret
    no shut
```

```
interface vlan 201
    ip address 10.0.2.3 255.255.255.0
    standby 200 ip 10.0.2.4
    standby 200 priority 110
    standby 200 preempt
    standby 200 timers 5 15
    standby 200 authentication Secret
    no shut
interface vlan 202
    ip address 10.0.3.3 255.255.255.0
    standby 200 ip 10.0.3.4
    standby 200 priority 110
    standby 200 preempt
    standby 200 timers 5 15
    standby 200 authentication Secret
    no shut
interface range gigabitethernet 2/1-3
    channel-group 2 mode on
    switchport trunk encapsulation dot1q
    no shut
...
```

Understanding the Command-Line Interface

This appendix includes the following topics, which describe how to use the command-line interface (CLI) on the Firewall Services Module (FWSM):

Note The CLI uses similar syntax and other conventions to the Cisco IOS CLI, but the FWSM operating system is not a version of Cisco IOS software. Do not assume that a Cisco IOS CLI command works or has the same function with the FWSM.

Command Prompts

When you are in the system configuration or in single context mode, the prompt begins with the host name:

```
FWSM
```

When you are within a context, the prompt begins with the host name followed by the context name:

```
FWSM/context
```

The prompt changes depending on the access mode:

- Unprivileged mode:

  ```
  FWSM>
  ```

  ```
  FWSM/context>
  ```

- Privileged mode, accessible by entering the **enable** command:

```
FWSM#

FWSM/context#
```

- Configuration mode, accessible by entering the **configure terminal** command:

```
FWSM(config)#

FWSM/context(config)#
```

- Subcommand mode, accessible when you enter a command that places you in a subcommand mode, such as **class** or **interface**:

```
FWSM(config-class)#

FWSM/context(config-if)#
```

Syntax Formatting

Command syntax descriptions use the following conventions:

Table C-1

Convention	Description
bold	Bold text indicates commands and keywords that you enter literally as shown.
italics	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
|	A vertical line indicates a choice within an optional or required set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical line indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical line indicate a required choice.
[x {y | z}]	Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical line within square brackets indicate a required choice within an optional element.

Abbreviating Commands

You can abbreviate most commands down to the fewest unique characters for a command; for example, you can enter **`wr t`** to view the configuration instead of entering the full command **`write terminal`**, or you can enter **`en`** to start privileged mode and **`con te`** to start configuration mode. In addition, you can enter **`0`** to represent **`0.0.0.0`**.

Command Line Editing

The FWSM uses the same command-line editing conventions as Cisco IOS software. You can view all previously entered commands with the **show history** command or individually with the up arrow or **^p** command. Once you have examined a previously entered command, you can move forward in the list with the down arrow or **^n** command. When you reach a command you wish to reuse, you can edit it or press the **Enter** key to start it. You can also delete the word to the left of the cursor with **^w**, or erase the line with **^u**.

The FWSM permits up to 512 characters in a command; additional characters are ignored.

Filtering Show Command Output

You can use the "pipe" operator (|) with any **show** command and include a filter option and filtering expression. The filtering is performed by matching each output line with a regular expression, similar to Cisco IOS software. By selecting different filter options you can include or exclude all output that matches the expression. You can also display all output beginning with the line that matches the expression.

The syntax for using filtering options with the **show** command is as follows:

show command | {**include** | **exclude** | **begin** | **grep** [**-v**]} *regexp*

In this command string, the first vertical bar (|) is the pipe operator and must be included in the command. This operator directs the output of the **show** command to the filter. In the syntax diagram, the other vertical bars (|) indicate alternative options and are not part of the command.

The **include** option includes all output lines that match the regular expression. The **grep** option without **-v** has the same effect. The **exclude** option excludes all output lines that match the regular expression. The **grep** option with **-v** has the same effect. The **begin** option shows all the output lines starting with the line that matches the regular expression.

Replace *regexp* with any Cisco IOS regular expression. The regular expression is not enclosed in quotes or double-quotes, so be careful with trailing white spaces, which will be taken as part of the regular expression.

When creating regular expressions, you can use any letter or number that you want to match. In addition, certain keyboard characters have special meaning when used in regular expressions. Table C-2 lists the keyboard characters that have special meaning.

Table C-2 Using Special Characters in Regular Expressions

Character Type	Character	Special Meaning
period	.	Matches any single character, including white space.
asterisk	*	Matches 0 or more sequences of the pattern.
plus sign	+	Matches 1 or more sequences of the pattern.
caret	^	Matches the beginning of the input string.
dollar sign	$	Matches the end of the input string.
underscore	_	Matches a comma (,), left brace ({), right brace (}), left parenthesis, right parenthesis, the beginning of the input string, the end of the input string, or a space.
brackets	[]	Designates a range of single-character patterns.

Table C-2 Using Special Characters in Regular Expressions (continued)

Character Type	Character	Special Meaning
hyphen	-	Separates the end points of a range.
parentheses	()	(Border Gateway Protocol (BGP) specific) Designates a group of characters as the name of a confederation.

Command Output Paging

On commands such as **help** or**?**, **show**, **show xlate**, or other commands that provide long listings, you can determine if the information displays a screen and pauses, or lets the command run to completion. The **pager** command lets you choose the number of lines to display before the More prompt appears.

When paging is enabled, the following prompt appears:

```
<--- More --->
```

The More prompt uses syntax similar to the UNIX **more** command:

- To view another screen, press the Space bar.
- To view the next line, press the **Enter** key.
- To return to the command line, press the **q** key.

Adding Comments

You can precede a line with a colon (:) to create a comment. However, the comment only appears in the command history buffer and not in the configuration. Therefore, you can view the comment with the **show history** command or by pressing an arrow key to retrieve a previous command, but because the comment is not in the configuration, the **write terminal** command does not display it.

Text Configuration Files

This section describes how to format a text configuration file that you can download to the FWSM, and includes the following topics:

To download the file, see the “Downloading a Text Configuration” section on page 16-6.

How Commands Correspond with Lines in the Text File

The text configuration file includes lines that correspond with the commands described in this guide and in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

In most cases, commands described in this guide are preceded by a CLI prompt. The prompt in the following example is "FWSM(config)#":

```
FWSM(config)# class gold
```

In the text configuration file you are not prompted to enter commands, so the prompt is omitted:

```
class gold
```

Subcommands

Subcommands appear indented under the main command when entered at the command line. Your text file lines do not need to be indented, as long as the subcommands appear directly following the main command. For example, the following unindented text is read the same as indented text:

```
class silver
limit all 5%
class gold
    limit all 10%
```

Automatic Text Entries

When you download a configuration to the FWSM, the FWSM inserts some lines automatically. For example, the FWSM inserts lines for default settings or for the time the configuration was modified. You do not need to enter these automatic entries when you create your text file.

Line Order

For the most part, commands can be in any order in the file. However, some lines, such as access control entries (ACEs), are processed in the order they appear, and the order can affect the function of the access control list (ACL). Other commands might also have order requirements. For example, you must enter the **nameif** command for an interface before you assign an IP address to it because many subsequent commands use the name of the interface. Also, subcommands must directly follow the main command.

Commands Not Included in the Text Configuration

Some commands do not insert lines in the configuration. For example, a runtime command such as **show config** does not have a corresponding line in the text file. Commands that you might expect to have entries but do not are noted in this guide, such as **activation key** or **mode multiple**.

Passwords

The login, enable, and user passwords are automatically encrypted before they are stored in the configuration. For example, the encrypted form of the password “letmein” might look like jMorNbK0514fadBh. You can copy the configuration passwords to another FWSM in their encrypted form, but you cannot unencrypt the passwords yourself.

If you enter an unencrypted password in a text file, the FWSM does not automatically encrypt them when you copy the configuration to the FWSM. The FWSM only encrypts them when you save the running configuration from the command line using the **copy running-config startup-config** or **write memory** command.

Multiple Security Context Files

For multiple security contexts, the entire configuration consists of multiple parts:

- The security context configurations
- The system configuration, which identifies basic settings for the FWSM, including a list of contexts
- The admin context, which provides network interfaces for the system configuration

  The system configuration does not include any interfaces or network settings for itself. Rather, when the system needs to access network resources (such as downloading the contexts from the server), it uses a context that is designated as the admin context.

Each context is similar to a single context mode configuration. The system configuration differs from a context configuration in that the system configuration includes system-only commands (such as a list of all contexts) while other typical commands are not present (such as many interface parameters).

See Chapter 5, “Managing Security Contexts,” for more information about contexts.

Command Help

Help information is available from the command line by entering **`help`** or a question mark to list all commands, or after a command to list command syntax; for example, **`arp ?`**.

The number of commands listed when you use the question mark or **help** command differs by access mode so that unprivileged mode offers the least commands and configuration mode offers the greatest number of commands.

In addition, you can enter any command by itself on the command line and then press **Enter** to view the command syntax.

Addresses, Protocols, and Ports Reference

This appendix provides a quick reference for the following categories:

IP Addresses and Subnet Masks

This section describes how to use IP addresses in the Firewall Services Module (FWSM). An IP address is a 32-bit number written in dotted decimal notation: four 8-bit fields (octets) converted from binary to decimal numbers, separated by dots. The first part of an IP address identifies the network on which the host resides, while the second part identifies the particular host on the given network. The network number field is called the network prefix. All hosts on a given network share the same network prefix but must have a unique host number. In classful IP, the class of the address determines the boundary between the network prefix and the host number.

This section includes the following topics:

Classes

IP host addresses are divided into three different address classes: Class A, Class B, and Class C. Each class fixes the boundary between the network prefix and the host number at a different point within the 32-bit address. Class D addresses are reserved for multicast IP.

- Class A addresses (1.xxx.xxx.xxx through 126.xxx.xxx.xxx) use only the first octet as the network prefix.
- Class B addresses (128.0.xxx.xxx through 191.255.xxx.xxx) use the first two octets as the network prefix.
- Class C addresses (192.0.0.xxx through 223.255.255.xxx) use the first three octets as the network prefix.

Because Class A addresses have 16,777,214 host addresses, and Class B addresses 65,534 hosts, you can use subnet masking to break these huge networks into smaller subnets.

Private Networks

If you need large numbers of addresses on your network, and they do not need to be routed on the Internet, you can use private IP addresses that the Internet Assigned Numbers Authority (IANA) recommends (see RFC 1918). The following address ranges are designated as private networks that should not be advertised:

- 10.0.0.0 through 10.255.255.255
- 172.16.0.0 through 172.31.255.255
- 192.168.0.0 through 192.168.255.255

Subnet Masks

A subnet mask lets you convert a single Class A, B, or C network into multiple networks. With a subnet mask, you can create an extended network prefix that adds bits from the host number to the network prefix. For example, a Class C network prefix always consists of the first three octets of the IP address. But a Class C extended network prefix uses part of the fourth octet as well.

Subnet masking is easy to understand if you use binary notation instead of dotted decimal. The bits in the subnet mask have a one-to-one correspondence with the Internet address:

- The bits are set to 1 if the corresponding bit in the IP address is part of the extended network prefix.
- The bits are set to 0 if the bit is part of the host number.

Example 1: If you have the Class B address 129.10.0.0 and you want to use the entire third octet as part of the extended network prefix instead of the host number, you must specify a subnet mask of 11111111.11111111.11111111.00000000. This subnet mask converts the Class B address into the equivalent of a Class C address, where the host number consists of the last octet only.

Example 2: If you want to use only part of the third octet for the extended network prefix, then you must specify a subnet mask like 11111111.11111111.11111000.00000000, which uses only 5 bits of the third octet for the extended network prefix.

You can write a subnet mask as a dotted decimal mask or as a */bits* (“slash *bits*”) mask. In Example 1, for a dotted decimal mask, you convert each binary octet into a decimal number: 255.255.255.0. For a */bits* mask, you add the number of 1s: /24. In Example 2, the decimal number is 255.255.248.0 and the /bits is /21.

You can also supernet multiple Class C networks into a larger network by using part of the third octet for the extended network prefix. For example, 192.168.0.0/20.

Determining the Subnet Mask

To determine the subnet mask based on how many hosts you want, see Table D-1.

Table D-1 Hosts, Bits, and Dotted Decimal Masks

Hosts[1]	/Bits Mask	Dotted Decimal Mask
16,777,216	/8	255.0.0.0 Class A Network
65,536	/16	255.255.0.0 Class B Network
32,768	/17	255.255.128.0
16,384	/18	255.255.192.0
8,192	/19	255.255.224.0
4,096	/20	255.255.240.0
2,048	/21	255.255.248.0
1,024	/22	255.255.252.0
512	/23	255.255.254.0
256	/24	255.255.255.0 Class C Network
128	/25	255.255.255.128
64	/26	255.255.255.192
32	/27	255.255.255.224
16	/28	255.255.255.240
8	/29	255.255.255.248
4	/30	255.255.255.252
Do not use	/31	255.255.255.254
1	/32	255.255.255.255 Single Host Address

1. The first and last number of a subnet are reserved, except for /32, which identifies a single host.

Determining the Address to Use with the Subnet Mask

The following sections describe how to determine the network address to use with a subnet mask for a Class C-size and a Class B-size network:

- Class C-Size Network Address, page D-4
- Class B-Size Network Address, page D-4

Class C-Size Network Address

For a network between 2 and 254 hosts, the fourth octet falls on a multiple of the number of host addresses, starting with 0. For example, the 8-host subnets (/29) of 192.168.0.x are as follows:

Subnet with Mask /29 (255.255.255.248)	Address Range[1]
192.168.0.0	192.168.0.0 to 192.168.0.7
192.168.0.8	192.168.0.8 to 192.168.0.15
192.168.0.16	192.168.0.16 to 192.168.0.31
…	…
192.168.0.248	192.168.0.248 to 192.168.0.255

1. The first and last address of a subnet are reserved. In the first subnet example, you cannot use 192.168.0.0 or 192.168.0.7.

Class B-Size Network Address

To determine the network address to use with the subnet mask for a network with between 254 and 65,534 hosts, you need to determine the value of the third octet for each possible extended network prefix. For example, you might want to subnet an address like 10.1.*x*.0, where the first two octets are fixed because they are used in the extended network prefix, and the fourth octet is 0 because all bits are used for the host number.

To determine the value of the third octet, follow these steps:

Step 1 Calculate how many subnets you can make from the network by dividing 65,536 (the total number of addresses using the third and fourth octet) by the number of host addresses you want.

For example, 65,536 divided by 4096 hosts equals 16.

Therefore, there are 16 subnets of 4096 addresses each in a Class B-size network.

Step 2 Determine the multiple of the third octet value by dividing 256 (the number of values for the third octet) by the number of subnets:

In this example, 256/16 = 16.

The third octet falls on a multiple of 16, starting with 0.

Therefore, the 16 subnets of the network 10.1 are as follows:

Subnet with Mask /20 (255.255.240.0)	Address Range[1]
10.1.0.0	10.1.0.0 to 10.1.15.255
10.1.16.0	10.1.16.0 to 10.1.31.255
10.1.32.0	10.1.32.0 to 10.1.47.255
…	…
10.1.240.0	10.1.240.0 to 10.1.255.255

1. The first and last address of a subnet are reserved. In the first subnet example, you cannot use 10.1.0.0 or 10.1.15.255.

Protocols and Applications

This section provides information about the protocols and applications with which you may need to work when configuring the FWSM. It includes the following topics:

Possible literal values are **ahp**, **eigrp**, **esp**, **gre**, **icmp**, **igmp**, **igrp**, **ip**, **ipinip**, **ipsec**, **nos**, **ospf**, **pcp**, **snp**, **tcp**, and **udp**. You can also specify any protocol by number. The **esp** and **ah** protocols only work in conjunction with Private Link.

Note The FWSM does not pass multicast packets. Many routing protocols use multicast packets for data transfer. If you need to send routing protocols across the FWSM, configure the routers with the Cisco IOS software **neighbor** command. We consider it inherently dangerous to send routing protocols across the FWSM. If the routes on the unprotected interface are corrupted, the routes transmitted to the protected side of the firewall will pollute routers there as well.

Table D-2 lists the numeric values for the protocol literals.

Table D-2 Protocol Literal Values

Literal	Value	Description
ah	51	Authentication Header for IPv6, RFC 1826
eigrp	88	Enhanced Interior Gateway Routing Protocol
esp	50	Encapsulated Security Payload for IPv6, RFC 1827
gre	47	generic routing encapsulation
icmp	1	Internet Control Message Protocol, RFC 792
igmp	2	Internet Group Management Protocol, RFC 1112
igrp	9	Interior Gateway Routing Protocol
ip	0	Internet Protocol
ipinip	4	IP-in-IP encapsulation
nos	94	Network Operating System (Novell's NetWare)
ospf	89	Open Shortest Path First routing protocol, RFC 1247
pcp	108	Payload Compression Protocol
snp	109	Sitara Networks Protocol
tcp	6	Transmission Control Protocol, RFC 793
udp	17	User Datagram Protocol, RFC 768

Protocol numbers can be viewed online at the IANA website:

http://www.iana.org/assignments/protocol-numbers

TCP and UDP Ports

Table D-3 lists the literal values and port numbers; either can be entered in FWSM commands. See the following caveats:

- The FWSM uses port 1521 for SQL*Net. This is the default port used by Oracle for SQL*Net. This value, however, does not agree with IANA port assignments.
- The FWSM listens for RADIUS on ports 1645 and 1646. If your RADIUS server uses the standard ports 1812 and 1813, you can configure the FWSM to listen to those ports using the **aaa-server radius-authport** and **aaa-server radius-acctport** commands.
- To assign a port for DNS access, use **domain**, not **dns**. The **dns** keyword translates into the port value for **dnsix**.

Port numbers can be viewed online at the IANA website:

http://www.iana.org/assignments/port-numbers

Table D-3 Port Literal Values

Literal	TCP or UDP?	Value	Description
aol	TCP	5190	America On-line
bgp	TCP	179	Border Gateway Protocol, RFC 1163
biff	UDP	512	Used by mail system to notify users that new mail is received
bootpc	UDP	68	Bootstrap Protocol Client
bootps	UDP	67	Bootstrap Protocol Server
chargen	TCP	19	Character Generator
citrix-ica	TCP	1494	Citrix Independent Computing Architecture (ICA) protocol
cmd	TCP	514	Similar to **exec** except that **cmd** has automatic authentication
ctiqbe	TCP	2748	Computer Telephony Interface Quick Buffer Encoding
daytime	TCP	13	Day time, RFC 867
discard	TCP, UDP	9	Discard
domain	TCP, UDP	53	DNS (Domain Name System)
dnsix	UDP	195	DNSIX Session Management Module Audit Redirector
echo	TCP, UDP	7	Echo
exec	TCP	512	Remote process execution
finger	TCP	79	Finger
ftp	TCP	21	File Transfer Protocol (control port)
ftp-data	TCP	20	File Transfer Protocol (data port)

Table D-3 Port Literal Values (continued)

Literal	TCP or UDP?	Value	Description
gopher	TCP	70	Gopher
https	TCP	443	Hyper Text Transfer Protocol (SSL)
h323	TCP	1720	H.323 call signalling
hostname	TCP	101	NIC Host Name Server
ident	TCP	113	Ident authentication service
imap4	TCP	143	Internet Message Access Protocol, version 4
irc	TCP	194	Internet Relay Chat protocol
isakmp	UDP	500	Internet Security Association and Key Management Protocol
kerberos	TCP, UDP	750	Kerberos
klogin	TCP	543	KLOGIN
kshell	TCP	544	Korn Shell
ldap	TCP	389	Lightweight Directory Access Protocol
ldaps	TCP	636	Lightweight Directory Access Protocol (SSL)
lpd	TCP	515	Line Printer Daemon - printer spooler
login	TCP	513	Remote login
lotusnotes	TCP	1352	IBM Lotus Notes
mobile-ip	UDP	434	MobileIP-Agent
nameserver	UDP	42	Host Name Server
netbios-ns	UDP	137	NetBIOS Name Service
netbios-dgm	UDP	138	NetBIOS Datagram Service
netbios-ssn	TCP	139	NetBIOS Session Service
nntp	TCP	119	Network News Transfer Protocol
ntp	UDP	123	Network Time Protocol
pcanywhere-status	UDP	5632	pcAnywhere status
pcanywhere-data	TCP	5631	pcAnywhere data
pim-auto-rp	TCP, UDP	496	Protocol Independent Multicast, reverse path flooding, dense mode
pop2	TCP	109	Post Office Protocol - Version 2
pop3	TCP	110	Post Office Protocol - Version 3
pptp	TCP	1723	Point-to-Point Tunneling Protocol
radius	UDP	1645	Remote Authentication Dial-In User Service
radius-acct	UDP	1646	Remote Authentication Dial-In User Service (accounting)

Table D-3 Port Literal Values (continued)

Literal	TCP or UDP?	Value	Description
rip	UDP	520	Routing Information Protocol
secureid-udp	UDP	5510	SecureID over UDP
smtp	TCP	25	Simple Mail Transport Protocol
snmp	UDP	161	Simple Network Management Protocol
snmptrap	UDP	162	Simple Network Management Protocol - Trap
sqlnet	TCP	1521	Structured Query Language Network
ssh	TCP	22	Secure Shell
sunrpc (rpc)	TCP, UDP	111	Sun Remote Procedure Call
syslog	UDP	514	System Log
tacacs	TCP, UDP	49	Terminal Access Controller Access Control System Plus
talk	TCP, UDP	517	Talk
telnet	TCP	23	RFC 854 Telnet
tftp	UDP	69	Trivial File Transfer Protocol
time	UDP	37	Time
uucp	TCP	540	UNIX-to-UNIX Copy Program
who	UDP	513	Who
whois	TCP	43	Who Is
www	TCP	80	World Wide Web
xdmcp	UDP	177	X Display Manager Control Protocol

ICMP Types

Table D-4 lists the ICMP type numbers and names that you can enter in FWSM commands:

Table D-4 ICMP Types

ICMP Number	ICMP Name
0	echo-reply
3	unreachable
4	source-quench
5	redirect
6	alternate-address
8	echo
9	router-advertisement
10	router-solicitation
11	time-exceeded
12	parameter-problem
13	timestamp-request
14	timestamp-reply
15	information-request
16	information-reply
17	mask-request
18	mask-reply
31	conversion-error
32	mobile-redirect

Acronyms and Abbreviations

This appendix lists the acronyms and abbreviations used in this document.

For more information on acronyms used in this guide, refer to the *Internetworking Terms and Acronyms* guide, which can be viewed online at the following website:

http://www.cisco.com/univercd/cc/td/doc/cisintwk/ita/index.htm

Table E-1 Acronyms and Abbreviations

Abbreviation	Description
AAA	authentication, authorization, and accounting.
ACE	Access Control Entry.
ACK	acknowledgement notification.
ACL	access control list.
AH	Authentication Header.
ARP	Address Resolution Protocol—A low-level TCP/IP protocol that maps a node's hardware address (called a "MAC" address) to its IP address. Defined in RFC 826. An example hardware address is 00:00:a6:00:01:ba. (The first three groups specify the manufacturer, the rest identify the host's motherboard.)
ASA	Adaptive Security Algorithm.
ASBR	Autonomous System Boundary Router.
ASCII	American Standard Code for Information Interchange.
BER	bit error rate.
BIND	Berkeley Internet Name Domain.
BGP	Border Gateway Protocol—While the Firewall Services Module (FWSM) does not support use of this protocol, you can set the routers on either side of the FWSM to use RIP between them and then run BGP on the rest of the network before the routers.
BOOTP	Bootstrap Protocol—Lets diskless workstations boot over the network and is described in RFC 951 and RFC 1542.
BPDU	bridge protocol data unit.
BSD	Berkeley Standard Distribution.
CA	certification authority.
CDP	Cisco Discovery Protocol.

Table E-1 Acronyms and Abbreviations (continued)

Abbreviation	Description
CGI	Common Gateway Interface.
chargen	Character Generation—Via TCP, a service that sends a continual stream of characters until stopped by the client. Via UDP, the server sends a random number of characters each time the client sends a datagram. Defined in RFC 864.
CLI	command-line interface.
conn	Connection slot in the FWSM—Refer to the **xlate** command page in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information.
CoS	Class of Service.
CPU	Central Processing Unit.
CR	carriage return.
CTIQBE	Computer Telephony Interface Quick Buffer Encoding.
DES	Data Encryption Standard.
DHCP	Dynamic Host Configuration Protocol.
DMZ	demilitarized zone—A separate network behind the firewall that allows limited access to outside users.
DNAT	Dynamic Network Address Translation.
DNS	Domain Name System—Operates over UDP unless zone file access over TCP is required.
DNS	Domain Name System (or Service).
DoS	Denial of service.
EIGRP	Enhanced Interior Gateway Routing Protocol—While the FWSM does not support use of this protocol, you can set the routers on either side of the FWSM to use RIP between them and then run EIGRP on the rest of the network before the routers.
EOBC	Ethernet Out-of-Band Channel.
ESP	Encapsulating Security Payload. Refer to RFC 1827 for more information.
EXEC	privileged command mode, which displays the “#” prompt.
Firewall MC	Firewall Management Center.
FTP	File Transfer Protocol.
FWSM	Firewall Services Module.
Gbps	Gigabit bytes per second.
GRE	Generic Routing Encapsulation—A tunneling protocol that does not use encryption.
H.323	A collection of protocols that allow the transmission of voice data over TCP/IP networks.
HTTP	HyperText Transfer Protocol—The service that handles access to the World Wide Web.
HTTPS	HTTP over SSL.

Table E-1 Acronyms and Abbreviations (continued)

Abbreviation	Description
IANA	Internet Assigned Number Authority—Assigns all port and protocol numbers for use on the Internet. You can view port numbers at the following site: http://www.iana.org/assignments/port-numbers You can view protocol numbers at the following site: http://www.iana.org/assignments/protocol-numbers
ICMP	Internet Control Message Protocol—This protocol is commonly used with the **ping** command. You can view ICMP traces through the FWSM with the **debug trace on** command. Refer to RFC 792 for more information.
IETF	Internet Engineering Task Force.
IGMP	Internet Group Management Protocol.
IGRP	Interior Gateway Routing Protocol.
IKE	Internet Key Exchange.
ILS	Internet Locator Service.
IOS	Internetwork Operating System.
IP	Internet Protocol.
IPinIP	IP-in-IP encapsulation protocol.
IPSec	IP Security Protocol efforts in the IETF (Internet Engineering Task Force).
IPX	Internetwork Packet Exchange.
IRC	Internet Relay Chat protocol—The protocol that lets users access chat rooms.
ISAKMP	Internet Security Association and Key Management Protocol.
ISC	IP Solution Center.
ISN	Initial Sequence Number.
ISP	Internet service provider.
ITU	International Telecommunication Union.
LDAP	Lightweight Directory Access Protocol.
LF	linefeed.
LSA	link-state advertisement.
MAC	Media Access Control.
MD5	Message Digest 5—An encryption standard for encrypting VPN packets. This same encryption is used with the **aaa authentication console** command to encrypt Telnet sessions to the console.
MGCP	Media Gateway Control Protocol.
MIB	Management Information Base—Used with SNMP.
MPLS	Multiprotocol Label Switching.
Mpps	Million packets per second.
MSFC	Multilayer Switch Feature Card.

Table E-1 Acronyms and Abbreviations (continued)

Abbreviation	Description
MTU	maximum transmission unit—The maximum number of bytes in a packet that can flow efficiently across the network with best response time. For Ethernet, the default MTU is 1500 bytes, but each network can have different values, with serial connections having the smallest values. The MTU is described in RFC 1191.
NAT	Network Address Translation.
NetBIOS	Network Basic Input Output System—An application programming interface (API) that provides special functions for PCs in local-area networks (LANs).
NFS	Network File System.
NIC	Network Information Center.
NIS	Network Information Service.
NMS	network management station.
NNTP	Network News Transfer Protocol—News reader service.
NOS	Network Operating System.
NP	Network Processor—as in IBM NP or Intel NP.
NSSA	not so stubby area.
NTP	Network Time Protocol—Set system clocks via the network.
OSPF	Open Shortest Path First.
PAT	Port Address Translation.
PBX	private branch exchange.
PDM	PDM for FWSM.
PDU	protocol data unit.
PIM	Protocol Independent Multicast.
PIX	Private Internet Exchange.
POP	Post Office Protocol.
PPP	Point-to-Point Protocol. Provides FWSM-to-router and host-to-network connections over synchronous and asynchronous circuits.
PPPoE	Point-to-Point Protocol over Ethernet.
PPTP	Point-to-Point Tunneling Protocol. RFC 2637 describes the PPTP protocol.
RADIUS	Remote Authentication Dial-In User Service—User authentication server specified with the **aaa-server** command.
RAS	The registration, admission, and status protocol. Provided with H.323 support.
RDT	Real Data Transport
RFC	Request For Comment—RFCs are the defacto standards of networking protocols.
RIP	Routing Information Protocol.
RPC	Remote Procedure Call.
RSA	Rivest, Shamir, and Adelman. RSA is the trade name for RSA Data Security, Inc.
RSH	Remote Shell—as in Remote Shell protocol.
RTCP	RTP Control Protocol.

Table E-1 Acronyms and Abbreviations (continued)

Abbreviation	Description
RTP	Real-Time Transport Protocol.
RTSP	Real Time Streaming Protocol.
SA	security association.
SCCP	Simple (Skinny) Client Control Protocol.
SCCP	Skinny (or Simple) Client Control Protocol is a simplified protocol used in VoIP networks.
SDP	Session Description Protocol.
SIP	Session Initiation Protocol.
SMTP	Simple Mail Transfer Protocol—Mail service. The **fixup protocol smtp** command enables the Mail Guard feature. The Mail Guard feature is compliant with both the RFC 1651 EHLO and RFC 821 section 4.5.1 commands.
SNMP	Simple Network Management Protocol—Set attributes with the **snmp-server** command.
SPC	Shared Profile Component.
SPF	shortest path first.
SPI	Security Parameter Index—A number which, together with a destination IP address and security protocol, uniquely identifies a particular security association.
SQL*Net	SQL*Net is a protocol Oracle uses to communicate between client and server processes. (SQL stands for Structured Query Language.)
SSH	Secure Shell.
SSH	Secure Shell.
STDERR	standard error file descriptor.
SVI	switched virtual interface.
SYN	Synchronize sequence numbers flag in the TCP header.
SYN	TCP synchronization, used as part of three-way handshake to establish a TCP session.
TACACS+	Terminal Access Controller Access Control System Plus.
TCP	Transmission Control Protocol. Refer to RFC 793 for more information.
TFTP	Trivial File Transfer Protocol.
TNSFrame	Transparent Network Substrate Frame.?
TPKT	Transport Packet.
Triple DES	Triple Data Encryption Standard. Also known as 3DES.
uauth	User authentication.
UDP	User Datagram Protocol.
URL	Universal Resource Locator.
UUIE	user-user information element.
VLAN	virtual LAN.
VoIP	Voice over IP.

Table E-1 Acronyms and Abbreviations (continued)

Abbreviation	Description
VPN	Virtual Private Network.
WAN	wide-area network.
WINS	Windows Internet Naming Service.
WWW	World Wide Web.
XDMCP	X Display Manager Control Protocol.
xlate	Translation session.

Symbols

A

B

C

D

E

F

G

H

L

M

N

O

P

Q

R

S

T

U

V

W

X