# Cisco MDS 9000 Family NX-OS Security Configuration Guide

First Published: Jan 28, 2016



**Cisco Systems, Inc.**
www.cisco.com

Cisco has more than 200 offices worldwide.
Addresses, phone numbers, and fax numbers
are listed on the Cisco website at
www.cisco.com/go/offices.

Text Part Number:

*Cisco MDS 9000 Family NX-OS Security Configuration Guide*

# CONTENTS

# New and Changed Information

Table 1 lists the New and Changed features for this guide

*Table 1 New and Changed Features*

| Feature | Release | Where Documented |
|---|---|---|
| Secure Login Enhancements | 7.3(1)DY(1) | Chapter 3, “Secure Login Enhancements” |
| SHA2 support for IPSec and IKEv2 on Cisco MDS 9700 Series Switches | 7.3(1)DY(1) | Chapter 7, “Configuring IPsec Network Security” |
| SHA2 support for IPSec and IKEv2 | 7.3(0)D1(1) | Chapter 7, “Configuring IPsec Network Security” |

# Preface

This preface describes the audience, organization, and conventions of the *Cisco MDS 9000 Family NX-OS Security Configuration Guide*. It also provides information on how to obtain related documentation.

## Audience

This guide is for experienced network administrators who are responsible for configuring and maintaining the Cisco MDS 9000 Family of multilayer directors and fabric switches.

## Document Organization

This document is organized as follows:

| | Title | Description |
|---|---|---|
| Chapter 1 | Security Overview | Provides an overview of the security features supported by the Cisco MDS 9000 Family NX-OS software. |
| Chapter 2 | Configuring FIPS | Describes the configuration guidelines for FIPS and also how to enable FIPS mode and how to conduct FIPS self-tests. |
| Chapter 5 | Configuring Users and Common Roles | Describes how to configure users and common roles. |
| Chapter 3 | Configuring Security Features on an External AAA Server | Describes the AAA parameters, user profiles, and RADIUS authentication security options provided in all switches in the Cisco MDS 9000 Family and provides configuration information for these options. |
| Chapter 4 | Configuring IPv4 and IPv6 Access Control Lists | Describes the IPv4 static routing feature and its use to route traffic between VSANs. |
| Chapter 6 | Configuring Certificate Authorities and Digital Certificates | Describes how to interoperate with Certificate Authorities (CAs) and use digital certificates for secure, scalable communication.**Chapter** |

| | Title | Description |
|---|---|---|
| Chapter 7 | Configuring IPsec Network Security | Provides details on the digital certificates, IP Security Protocol (IPsec) open standards, and the Internet Key Exchange (IKE) protocol that it uses to handle protocol and algorithm negotiation. |
| Chapter 8 | Configuring FC-SP and DHCHAP | Describes the DHCHAP protocol, an FC-SP protocol, that provides authentication between Cisco MDS 9000 Family switches and other devices. |
| Chapter 9 | Configuring Port Security | Provides details on port security features that can prevent unauthorized access to a switch port in the Cisco MDS 9000 Family. |
| Chapter 10 | Configuring Fibre Channel Common Transport Management Security | Provides details on how to configure the Fire Channel Transport Management server query so that only a network administrator can send queries to a switch and access information. |
| Chapter 11 | Configuring Fabric Binding | Describes the fabric binding security feature for VSANs, which ensures that ISLs are only enabled between specific switches. |
| Chapter 12 | Configuring Cisco TrustSec Fibre Channel Link Encryption | Describes how the switch allows IP hosts to access Fibre Channel storage using the iSCSI protocol. |

# Document Conventions

Command descriptions use these conventions:

| | |
|---|---|
| **boldface font** | Commands and keywords are in boldface. |
| *italic font* | Arguments for which you supply values are in italics. |
| **[ ]** | Elements in square brackets are optional. |
| **[ x \| y \| z ]** | Optional alternative keywords are grouped in brackets and separated by vertical bars. |

Screen examples use these conventions:

| | |
|---|---|
| `screen font` | Terminal sessions and information the switch displays are in screen font. |
| **`boldface screen font`** | Information you must enter is in boldface screen font. |
| *italic screen font* | Arguments for which you supply values are in italic screen font. |
| `< >` | Nonprinting characters, such as passwords, are in angle brackets. |
| `[ ]` | Default responses to system prompts are in square brackets. |
| `!, #` | An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line. |

This document uses the following conventions:

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

**Caution** Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

# Related Documentation

The documentation set for the Cisco MDS 9000 Family includes the following documents. To find a document online, use the Cisco MDS NX-OS Documentation Locator at:

http://www.cisco.com/en/US/docs/storage/san_switches/mds9000/roadmaps/doclocater.htm

## Release Notes

- *Cisco MDS 9000 Family Release Notes for Cisco MDS NX-OS Releases*
- *Cisco MDS 9000 Family Release Notes for MDS SAN-OS Releases*
- *Cisco MDS 9000 Family Release Notes for Storage Services Interface Images*
- *Cisco MDS 9000 Family Release Notes for Cisco MDS 9000 EPLD Images*
- *Release Notes for Cisco MDS 9000 Family Fabric Manager*

## Regulatory Compliance and Safety Information

- *Regulatory Compliance and Safety Information for the Cisco MDS 9000 Family*

## Compatibility Information

- *Cisco Data Center Interoperability Support Matrix*
- *Cisco MDS 9000 NX-OS Hardware and Software Compatibility Information and Feature Lists*
- *Cisco MDS NX-OS Release Compatibility Matrix for Storage Service Interface Images*
- *Cisco MDS 9000 Family Switch-to-Switch Interoperability Configuration Guide*
- *Cisco MDS NX-OS Release Compatibility Matrix for IBM SAN Volume Controller Software for Cisco MDS 9000*
- *Cisco MDS SAN-OS Release Compatibility Matrix for VERITAS Storage Foundation for Networks Software*

# Hardware Installation

- *Cisco MDS 9710 Series Hardware Installation Guide*
- *Cisco MDS 9500 Series Hardware Installation Guide*
- *Cisco MDS 9250i Series Hardware Installation Guide*
- *Cisco MDS 9200 Series Hardware Installation Guide*
- *Cisco MDS 9100 Series Hardware Installation Guide*
- *Cisco MDS 9148S Series Hardware Installation Guide*
- *Cisco MDS 9148S Multilayer Fabric Switch Quick Start Guide*
- *Cisco MDS 9124 and Cisco MDS 9134 Multilayer Fabric Switch Quick Start Guide*

# Software Installation and Upgrade

- *Cisco MDS 9000 NX-OS Release 4.1(x) and SAN-OS 3(x) Software Upgrade and Downgrade Guide*
- *Cisco MDS 9000 Family Storage Services Interface Image Install and Upgrade Guide*
- *Cisco MDS 9000 Family Storage Services Module Software Installation and Upgrade Guide*

# Cisco NX-OS

- *Cisco MDS 9000 Family NX-OS Licensing Guide*
- *Cisco MDS 9000 Family NX-OS Fundamentals Configuration Guide*
- *Cisco MDS 9000 Family NX-OS System Management Configuration Guide*
- *Cisco MDS 9000 Family NX-OS Interfaces Configuration Guide*
- *Cisco MDS 9000 Family NX-OS Fabric Configuration Guide*
- *Cisco MDS 9000 Family NX-OS Quality of Service Configuration Guide*
- *Cisco MDS 9000 Family NX-OS Security Configuration Guide*
- *Cisco MDS 9000 Family NX-OS IP Services Configuration Guide*
- *Cisco MDS 9000 Family NX-OS Intelligent Storage Services Configuration Guide*
- *Cisco MDS 9000 Family NX-OS High Availability and Redundancy Configuration Guide*
- *Cisco MDS 9000 Family NX-OS Inter-VSAN Routing Configuration Guide*

# Cisco Fabric Manager

- *Cisco Fabric Manager Fundamentals Configuration Guide*
- *Cisco Fabric Manager System Management Configuration Guide*
- *Cisco Fabric Manager Interfaces Configuration Guide*
- *Cisco Fabric Manager Fabric Configuration Guide*
- *Cisco Fabric Manager Quality of Service Configuration Guide*
- *Cisco Fabric Manager Security Configuration Guide*

- *Cisco Fabric Manager IP Services Configuration Guide*
- *Cisco Fabric Manager Intelligent Storage Services Configuration Guide*
- *Cisco Fabric Manager High Availability and Redundancy Configuration Guide*
- *Cisco Fabric Manager Inter-VSAN Routing Configuration Guide*
- *Cisco Fabric Manager Online Help*
- *Cisco Fabric Manager Web Services Online Help*

## Command-Line Interface

- *Cisco MDS 9000 Family Command Reference*

## Intelligent Storage Networking Services Configuration Guides

- *Cisco MDS 9000 I/O Acceleration Configuration Guide*
- *Cisco MDS 9000 Family SANTap Deployment Guide*
- *Cisco MDS 9000 Family Data Mobility Manager Configuration Guide*
- *Cisco MDS 9000 Family Storage Media Encryption Configuration Guide*
- *Cisco MDS 9000 Family Secure Erase Configuration Guide*
- *Cisco MDS 9000 Family Cookbook for Cisco MDS SAN-OS*

## Troubleshooting and Reference

- *Cisco NX-OS System Messages Reference*
- *Cisco MDS 9000 Family NX-OS Troubleshooting Guide*
- *Cisco MDS 9000 Family NX-OS MIB Quick Reference*
- *Cisco MDS 9000 Family NX-OS SMI-S Programming Reference*
- *Cisco MDS 9000 Family Fabric Manager Server Database Schema*

# Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as an RSS feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service. Cisco currently supports RSS Version 2.0.

CHAPTER 1

# Security Overview

The Cisco MDS 9000 NX-OS software supports advanced security features that provide security within a Storage Area Network (SAN). These features protect your network against deliberate or unintentional disruptions from internal or external threats.

This chapter includes the following sections:



## FIPS

The Federal Information Processing Standards (FIPS) Publication 140-2, *Security Requirements for Cryptographic Modules*, details the U.S. government requirements for cryptographic modules. FIPS 140-2 specifies that a cryptographic module should be a set of hardware, software, firmware, or some combination that implements cryptographic functions or processes, including cryptographic algorithms and, optionally, key generation, and is contained within a defined cryptographic boundary. FIPS specifies certain crypto algorithms as secure, and it also identifies which algorithms should be used if a cryptographic module is to be called FIPS compliant.

For more information on configuring FIPS, see Chapter 2, “Configuring FIPS.”

# Users and Common Roles

Role-based authorization limits access to switch operations by assigning users to roles. All management access within the Cisco MDS 9000 Family is based upon roles. Users are restricted to performing the management operations that are explicitly permitted, by the roles to which they belong.

For information on configuring users and common roles, see Chapter 3, "Configuring Common Roles."

# RADIUS and TACACS+

The authentication, authorization, and accounting (AAA) feature verifies the identity of, grants access to, and tracks the actions of users managing a switch. All Cisco MDS 9000 Family switches use RADIUS and TACACS+ protocols to provide solutions using remote AAA servers. This security feature provides a centralized user account management capability for AAA servers.

AAA uses security protocols to administer its security functions. If your router or access server is acting as a network access server, then the communication between your network access server and the RADIUS or TACACS+ security server is through AAA.

The chapters in this guide describe the following features:

- Switch management—A management security system that provides security to all management access methods, including the command-line interface (CLI) or Simple Network Management Protocol (SNMP).
- Switch AAA functionalities—A function by which you can configure AAA switch functionalities on any switch in the Cisco MDS 9000 Family, using the command-line interface (CLI) or Simple Network Management Protocol (SNMP).
- RADIUS—A distributed client and server system implemented through AAA that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.
- TACACS+—A security application implemented through AAA that provides a centralized validation of users who are attempting to gain access to a router or network access server. TACACS+ services are maintained in a database on a TACACS+ daemon that typically runs on a UNIX or Windows NT workstation. TACACS+ provides for separate and modular authentication, authorization, and accounting facilities.

For information on configuring RADIUS and TACACS+, see Chapter 4, "Configuring Security Features on an External AAA Server"

# IP ACLs

IP access control lists (ACLs) provide basic network security on the out-of-band management Ethernet interface and the in-band IP management Interface. The Cisco MDS 9000 Family switches use IP ACLs to restrict traffic from unknown and untrusted sources and restrict network use based on user identity or device type.

For information on configuring IP ACLs, see Chapter 5, "Configuring IPv4 and IPv6 Access Control Lists"

# PKI

The Public Key Infrastructure (PKI) allows an MDS 9000 switch to obtain and use digital certificates for secure communication in the network. PKI support provides manageability and scalability for applications, such as IPsec, IKE, and SSH, that support digital certificates.

For information on configuring PKI, see Chapter 6, “Configuring Certificate Authorities and Digital Certificates.”

# IPsec

IP Security (IPsec) protocol is a framework of open standards by the Internet Engineering Task Force (IETF) that provides data confidentiality, data integrity, and data origin authentication between participating peers. IPsec provides security services at the IP layer, including protecting one or more data flows between a pair of hosts, a pair of security gateways, or a security gateway and a host.

For information on configuring IPsec, see Chapter 7, “Configuring IPsec Network Security.”

# FC-SP and DHCHAP

Fibre Channel Security Protocol (FC-SP) capabilities provide switch to switch and hosts to switch authentication to overcome security challenges for enterprise-wide fabrics. Diffie-Hellman Challenge Handshake Authentication Protocol (DHCHAP) is an FC-SP protocol that provides authentication between Cisco MDS 9000 Family switches and other devices. DHCHAP consists of the CHAP protocol combined with the Diffie-Hellman exchange.

With FC-SP, switches, storage devices, and hosts are able to prove their identity through a reliable and manageable authentication mechanism. With FC-SP, Fibre Channel traffic can be secured on a frame-by-frame basis to prevent snooping and hijacking, even over untrusted links. A consistent set of policies and management actions are propagated through the fabric to provide a uniform level of security across the entire fabric.

For more information on configuring FS-SP and DHCHAP, see Chapter 8, “Configuring FC-SP and DHCHAP.”

# Port Security

The port security feature prevents unauthorized access to a switch port by binding specific world-wide names (WWNs) that have access to one or more given switch ports.

When port security is enabled on a switch port, all devices connecting to that port must be in the port security database and must be listed in the database as bound to a given port. If both of these criteria are not met, the port will not achieve an operationally active state and the devices connected to the port will be denied access to the SAN.

For information on configuring port security, see Chapter 9, “Configuring Port Security.”

# Fibre Channel Common Transport Management Server Query

With the FC-CT query management feature, an administrator can configure the network in such a manner that only a storage administrator or a network administrator can send queries to a switch and access information such as devices that are logged in devices in the fabric, switches in the fabric, how they are connected, how many ports each switch has and where each port is connected, configured zone information and privilege to add or delete zone and zone sets, and Host Bus Adapter (HBA) details of all the hosts connected in the fabric and so on.

For information on configuring fabric binding, see Chapter 10, "Configuring Fibre Channel Common Transport Management Security."

# Fabric Binding

The fabric binding feature ensures Inter-Switch Links (ISLs) are enabled only between specified switches in the fabric binding configuration. This feature helps prevent unauthorized switches from joining the fabric or disrupting the current fabric operations. This feature uses the Exchange Fabric Membership Data (EEMD) protocol to ensure that the list of authorized switches is identical in all of the switches in a fabric.

For information on configuring fabric binding, see Chapter 11, "Configuring Fabric Binding."

# TrustSec Fibre Channel Link Encryption

Cisco TrustSec Fibre Channel Link Encryption is an extension of the Fibre Channel-Security Protocol (FC-SP) feature and uses the existing FC-SP architecture to provide integrity and confidentiality of transactions. Encryption is added to the peer authentication capability to provide security and prevent unwanted traffic interception. Peer authentication is implemented according to the FC-SP standard using the Diffie-Hellman Challenge Handshake Authentication Protocol (DHCHAP) protocol.

For information on configuring TrustSec Fibre Channel Link Encryption, see Chapter 12, "Configuring Cisco TrustSec Fibre Channel Link Encryption."

# Open IP Ports on Cisco MDS 9000 Series Platforms

Cisco MDS 9000 Series platforms with default configurations have IP ports that are open on the external management interface. The table below lists the open ports and their corresponding services:

*Table 1-1 Open IP Ports on Cisco MDS 9000 Series Platforms*

| Port number | IP Protocol (UDP/TCP) | Platform | Feature/Service Name | Random Port? |
|---|---|---|---|---|
| None | UDP | All | — | — |
| 600 - 1024 | TCP | All | NFS | Yes |
| 2002 | TCP | All | Remote Packet Capture | No |

*Table 1-1 Open IP Ports on Cisco MDS 9000 Series Platforms*

| Port number | IP Protocol (UDP/TCP) | Platform | Feature/Service Name | Random Port? |
|---|---|---|---|---|
| 7546 | TCP | All | CFS over IPv4 | No |
| 9333 | TCP | All | Cluster | No |
| 32768 - 32769 | TCP | Cisco MDS 8-Gb Fabric Switch for HP c-Class Blade System<br>Cisco MDS 9148<br>Cisco MDS 9222i<br>Cisco MDS 9506<br>Cisco MDS 9509<br>Cisco MDS 9513 | License Manager | Yes |
| 44583 - 59121 | TCP | Cisco MDS 9148S<br>Cisco MDS 9250i<br>Cisco MDS 9706<br>Cisco MDS 9710 | License Manager | Yes |

**NFS**—A port in this range is used by the NFS service on the switch. This is only for intraswitch use. It is not essential to provide external access to or from these ports. This feature cannot be disabled. To block access to this service, configure an IP access list to deny access to the range of ports. Refer to the Configuring IPv4 and IPv6 Access Control Lists section of the Cisco MDS 9000 Family NX-OS Security Configuration Guide for details.

**Remote Packet Capture**—This port is used by the Fibre Channel Analyzer service on the switch for communicating with an Ethereal protocol analyzer client on a host using the Remote Capture Protocol (RPCAP). This service is used for troubleshooting and is optional for normal switch operation. This feature cannot be disabled. To block access to this service, configure an IP access list to deny access to the range of ports. Refer to the Configuring IPv4 and IPv6 Access Control Lists section of the Cisco MDS 9000 Family NX-OS Security Configuration Guide for details.

**CFS over IPv4**—This port is used by the CFS over IPv4 service to distribute switch configuration information to peer switches in the fabric. CFS is an important service for a switch to communicate with peers, but several transport options are possible. The correct transport depends on the fabric implementation. This port may be closed by disabling the CFS over IPv4 service. Refer to the Enabling CFS Over IP section of the Cisco MDS 9000 Family CLI Configuration Guide for details.

**Cluster**—This port is used by the cluster service to communicate with peer switches in a cluster. Features such as IOA and SME rely on this service. If such features are not in use, the cluster service is not essential to a switch operation. This port can be closed by disabling the cluster service. Refer to the Enabling and Disabling Clustering section of the Cisco MDS 9000 Family Storage Media Encryption Configuration Guide for details.

**License Manager**—These ports are used by the License Manager service. This only for intraswitch use. It is not essential to provide external access to or from these ports. This feature cannot be disabled. To block access to this service, configure an IP access list to deny access to the range of ports. Refer to the Configuring IPv4 and IPv6 Access Control Lists section of the Cisco MDS 9000 Family NX-OS Security Configuration Guide for details.


# Configuring FIPS

The Federal Information Processing Standards (FIPS) Publication 140-2, *Security Requirements for Cryptographic Modules*, details the U.S. government requirements for cryptographic modules. FIPS 140-2 specifies that a cryptographic module should be a set of hardware, software, firmware, or some combination that implements cryptographic functions or processes, including cryptographic algorithms and, optionally, key generation, and is contained within a defined cryptographic boundary.

FIPS specifies certain crypto algorithms as secure, and it also identifies which algorithms should be used if a cryptographic module is to be called FIPS compliant.

**Note** Cisco MDS SAN-OS Release 3.1(1) and NX-OS Release 4.1(1b) or later implements FIPS features and is currently in the certification process with the U.S. government, but it is not FIPS compliant at this time.

This chapter includes the following sections:



## Configuration Guidelines

Follow these guidelines before enabling FIPS mode:

- Make your passwords a minimum of eight characters in length.
- Disable Telnet. Users should log in using SSH only.
- Disable remote authentication through RADIUS/TACACS+. Only users local to the switch can be authenticated.
- Disable SNMP v1 and v2. Any existing user accounts on the switch that have been configured for SNMPv3 should be configured only with SHA for authentication and AES/3DES for privacy.
- Disable VRRP.
- Delete all IKE policies that either have MD5 for authentication or DES for encryption. Modify the policies so they use SHA for authentication and 3DES/AES for encryption.
- Delete all SSH Server RSA1 keypairs.

# Enabling FIPS Mode

To enable FIPS mode, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# fips mode enable` | Enables FIPS mode. |
| | `switch(config)# no fips mode enable` | Disables FIPS mode. |

# Displaying FIPS Status

To view FIPS status, enter the **show fips status** command.

# FIPS Self-Tests

A cryptographic module must perform power-up self-tests and conditional self-tests to ensure that it is functional.

**Note** FIPS power-up self-tests automatically run when FIPS mode is enabled by entering the **fips mode enable** command. A switch is in FIPS mode only after all self-tests are successfully completed. If any of the self-tests fail, then the switch is rebooted.

Power-up self-tests run immediately after FIPS mode is enabled. A cryptographic algorithm test using a known answer must be run for all cryptographic functions for each FIPS 140-2-approved cryptographic algorithm implemented on the Cisco MDS 9000 Family.

Using a known-answer test (KAT), a cryptographic algorithm is run on data for which the correct output is already known, and then the calculated output is compared to the previously generated output. If the calculated output does not equal the known answer, the known-answer test fails.

Conditional self-tests must be run when an applicable security function or operation is invoked. Unlike the power-up self-tests, conditional self-tests are executed each time their associated function is accessed.

Conditional self-tests include the following:

- Pair-wise consistency test—This test is run when a public-private keypair is generated.
- Continuous random number generator test—This test is run when a random number is generated.

Both of these tests automatically run when a switch is in FIPS mode.


# Configuring Users and Common Roles

The CLI and SNMP use common roles in all switches in the Cisco MDS 9000 Family. You can use the CLI to modify a role that was created using SNMP and vice versa.

Users, passwords, and roles for all CLI and SNMP users are the same. A user configured through the CLI can access the switch using SNMP (for example, the Fabric Manager or the Device Manager) and vice versa.

This chapter includes the following sections:



## Feature Information

This section briefly describes the new and updated features for releases.

*Table 3-1 New and Changed Features*

| Feature | Release | Description |
|---|---|---|
| Secure Login Enhancements | 7.3(1)DY(1) | This feature allows users to enhance the security of Cisco MDS Switches by automatically blocking login attempts when a possible denial-of-service (DoS) attack is detected. |

# Role-Based Authorization

Switches in the Cisco MDS 9000 Family perform authentication based on roles. Role-based authorization limits access to switch operations by assigning users to roles. This kind of authentication restricts you to management operations based on the roles to which you have been added.

When you execute a command, perform command completion, or obtain context sensitive help, the switch software allows the operation to progress if you have permission to access that command.

This section includes the following topics:



## About Roles

Each role can contain multiple users and each user can be part of multiple roles. For example, if role1 users are only allowed access to configuration commands, and role2 users are only allowed access to **debug** commands, then if Joe belongs to both role1 and role2, he can access configuration as well as **debug** commands.

**Note** If you belong to multiple roles, you can execute a union of all the commands permitted by these roles. Access to a command takes priority over being denied access to a command. For example, suppose you belong to a TechDocs group and you were denied access to configuration commands. However, you also belong to the engineering group and have access to configuration commands. In this case, you will have access to configuration commands.

**Tip** Any role, when created, does not allow access to the required commands immediately. The administrator must configure appropriate rules for each role to allow access to the required commands.

## Configuring Roles and Profiles

To create an additional role or to modify the profile for an existing role, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# role name techdocs`<br>`switch(config-role)#` | Places you in the mode for the specified role (techdocs).<br>**Note** The role submode prompt indicates that you are now in the role submode. This submode is now specific to the techdocs group. |
| | `switch(config)# no role name techdocs` | Deletes the role called techdocs. |

| | Command | Purpose |
|---|---|---|
| Step 3 | `switch(config-role)# description Entire Tech Docs group` | Assigns a description to the new role. The description is limited to one line and can contain spaces. |
| | `switch(config-role)# no description` | Resets the description for the Tech Docs group. |

**Note** Only users belonging to the network-admin role can create roles.

# Configuring Rules and Features for Each Role

Up to 16 rules can be configured for each role. The user-specified rule number determines the order in which the rules are applied. For example, rule 1 is applied before rule 2, which is applied before rule 3, and so on. A user not belonging to the network-admin role cannot perform commands related to roles.

For example, if user A is permitted to perform all **show** commands, user A cannot view the output of the **show role** command if user A does not belong to the network-admin role.

The **rule** command specifies operations that can be performed by a specific role. Each rule consists of a rule number, a rule type (permit or deny), a command type (for example, **config**, **clear**, **show**, **exec**, **debug**), and an optional feature name (for example, FSPF, zone, VSAN, fcping, or interface).

**Note** In this case, **exec** commands refer to all commands in the EXEC mode that are not included in the **show**, **debug**, and **clear** command categories.

In cases where a default role is applicable to all users, and a configured role is applicable for specific users, consider the following scenarios:

- Same rule type (permit or deny)–If the default role and the configured role for a specific user have the same rule type, then the specific user will have access to all the rules of both the default role and the configured role.

  If the default role, say **A**, has the following rules:

  ```
  rule 5 permit show feature environment
  rule 4 permit show feature hardware
  rule 3 permit config feature ssh
  rule 2 permit config feature ntp
  rule 1 permit config feature tacacs+
  ```

  And, a specific user is assigned to the following role, say **B**, with one rule:

  ```
  rule 1 permit config feature dpvm
  ```

  The specific user will have access to the rules of both **A** and **B**.

- Different rule type–If the default role and the configured role for a specific user have different rule types for a particular rule, then the default role will override the conflicting rule statement of the configured role.

  If the default role, say **A**, has the following rules:

  ```
  rule 5 permit show feature environment
  rule 4 permit show feature hardware
  rule 3 permit config feature ssh
  rule 2 permit config feature ntp
  rule 1 permit config feature tacacs+
  ```

And, a specific user is assigned to the following role, say **B**, with two rules:

```
rule 6 permit config feature dpvm
rule 2 deny config feature ntp
```

Rule 2 of **A** and **B** are in conflict. In this case, **A** overrides the conflicting rule **of B**, and the user is assigned with the remaining rules of **A** and **B**, including the overridden rule:

```
rule 6 permit config feature dpvm
rule 5 permit show feature environment
rule 4 permit show feature hardware
rule 3 permit config feature ssh
rule 2 permit config feature ntp ----------> Overridden rule
rule 1 permit config feature tacacs+
```

## Rule Changes Between SAN-OS Release 3.3(1c) and NX-OS Release 4.2(1a) Affect Role Behavior

The rules that can be configured for roles were modified between SAN-OS Release 3.3(1c) and NX-OS Release 4.2(1a). As a result, roles do not behave as expected following an upgrade from SAN-OS Release 3.3(1c) to NX-OS Release 4.2(1a). Manual configuration changes are required to restore the desired behavior.

**Rule 4 and Rule 3**: after the upgrade, **exec** and **feature** are removed. Change rule 4 and rule 3 as follows:

| SAN-OS Release 3.3(1c) Rule | NX-OS Release 4.2(1a), Set the Rule to: |
|---|---|
| rule 4 permit exec feature debug | rule 4 permit debug |
| rule 3 permit exec feature clear | rule 3 permit clear |

**Rule 2**: after the upgrade, **exec feature license** is obsolete.

| SAN-OS Release 3.3(1c) Rule | NX-OS Release 4.2(1a) Rule |
|---|---|
| rule 2 permit exec feature debug | Not available in Release 4.2(1). |

**Rule 9, Rule 8, and Rule 7**: after the upgrade, you need to have the feature enabled to configure it. In SAN-OS Release 3.3(1c), you could configure a feature without enabling it.

| SAN-OS Release 3.3(1c) Rule | NX-OS Release 4.2(1a), to Preserve the Rule: |
|---|---|
| rule 9 deny config feature telnet | Not available in Release 4.2(1) and cannot be used. |
| rule 8 deny config feature tacacs-server | During the upgrade, enable the feature to preserve the rule; otherwise, the rule disappears. |
| rule 7 deny config feature tacacs+ | During the upgrade, enable the feature to preserve the rule; otherwise, the rule disappears. |

## Modifying Profiles

To modify the profile for an existing role, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# role name sangroup`<br>`switch(config-role)#` | Places you in role configuration submode for the existing role sangroup. |
| **Step 3** | `switch(config-role)# rule 1 permit config`<br>`switch(config-role)# rule 2 deny config feature fspf`<br>`switch(config-role)# rule 3 permit debug feature zone`<br>`switch(config-role)# rule 4 permit exec feature fcping` | Allows users belonging to the sangroup role to perform all configuration commands except **fspf config** commands. They can also perform **zone debug** commands and the **fcping** EXEC mode command. |
| **Step 4** | `switch(config-role)# no rule 4` | Deletes rule 4, which no longer permits the sangroup to perform the **fcping** command. |

In Step 3, rule 1 is applied first, thus permitting sangroup users access to all **config** commands. Rule 2 is applied next, denying FSPF configuration to sangroup users. As a result, sangroup users can perform all other **config** commands, except **fspf** configuration commands.

**Note** The order of rule placement is important. If you had swapped these two rules and issued the **deny config feature fspf** rule first and issued the **permit config** rule next, you would be allowing all sangroup users to perform all configuration commands because the second rule globally overrode the first rule.

# Configuring the VSAN Policy

Configuring the VSAN policy requires the ENTERPRISE_PKG license (for more information, see the *Cisco MDS 9000 Family NX-OS Licensing Guide*).

You can configure a role so that it only allows tasks to be performed for a selected set of VSANs. By default, the VSAN policy for any role is permit, which allows tasks to be performed for all VSANs. You can configure a role that only allows tasks to be performed for a selected set of VSANs. To selectively allow VSANs for a role, set the VSAN policy to deny, and then set the configuration to permit or the appropriate VSANs.

**Note** Users configured in roles where the VSAN policy is set to deny cannot modify the configuration for E ports. They can only modify the configuration for F or FL ports (depending on whether the configured rules allow such configuration to be made). This is to prevent such users from modifying configurations that may impact the core topology of the fabric.

**Tip** Roles can be used to create VSAN administrators. Depending on the configured rules, these VSAN administrators can configure MDS features (for example, zone, fcdomain, or VSAN properties) for their VSANs without affecting other VSANs. Also, if the role permits operations in multiple VSANs, then the VSAN administrators can change VSAN membership of F or FL ports among these VSANs.

Users belonging to roles in which the VSAN policy is set to deny are referred to as VSAN-restricted users.

## Modifying the VSAN Policy

**Note** Beginning with NX-OS Release 4.x, the VSAN enforcement is done only for non-show commands. The show commands are excluded.

**Note** In SAN-OS Release 3.x and lower, the VSAN enforcement is done for non-show commands, but, not all the show commands are enforced.

To modify the VSAN policy for an existing role, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# role name sangroup`<br>`switch(config-role)#` | Places you in role configuration submode for the sangroup role. |
| **Step 3** | `switch(config)# vsan policy deny`<br>`switch(config-role-vsan)` | Changes the VSAN policy of this role to **deny** and places you in a submode where VSANs can be selectively permitted. |
| | `switch(config-role)# no vsan policy deny` | Deletes the configured VSAN role policy and reverts to the factory default (**permit**). |
| **Step 4** | `switch(config-role-vsan)# permit vsan 10-30` | Permits this role to perform the allowed commands for VSANs 10 through 30. |
| | `switch(config-role-vsan)# no permit vsan 15-20` | Removes the permission for this role to perform commands for VSANs 15 to 20. So, the role is now permitted to perform commands for VSAN 10 to 14, and 21 to 30. |

# Role Distributions

Role-based configurations use the Cisco Fabric Services (CFS) infrastructure to enable efficient database management and to provide a single point of configuration for the entire fabric.

The following configurations are distributed:

- Role names and descriptions
- List of rules for the roles
- VSAN policy and the list of permitted VSANs

This section includes the following topics:

# About Role Databases

Role-based configurations use two databases to accept and implement configurations.

- Configuration database—The database currently enforced by the fabric.
- Pending database—Your subsequent configuration changes are stored in the pending database. If you modify the configuration, you need to commit or discard the pending database changes to the configuration database. The fabric remains locked during this period. Changes to the pending database are not reflected in the configuration database until you commit the changes.

**Note** As soon as the customer encounters syslog"%VSHD-4-VSHD_ROLE_DATABASE_OUT_OF_SYNC", Role configuration database is found to be different between the switches during merge. Role configuration database is recommended to be identical among all switches in the fabric. Edit the configuration on one of the switches to obtain the desired role configuration database and then commit it.

# Locking the Fabric

The first action that modifies the database creates the pending database and locks the feature in the entire fabric. Once you lock the fabric, the following situations apply:

- No other user can make any configuration changes to this feature.
- A copy of the configuration database becomes the pending database along with the first change.

# Committing Role-Based Configuration Changes

If you commit the changes made to the pending database, the configuration is committed to all the switches in the fabric. On a successful commit, the configuration change is applied throughout the fabric and the lock is released. The configuration database now contains the committed changes and the pending database is now cleared.

To commit role-based configuration changes, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# role commit vsan 3` | Commits the role-based configuration changes. |

# Discarding Role-Based Configuration Changes

If you discard (abort) the changes made to the pending database, the configuration database remains unaffected and the lock is released.

To discard role-based configuration changes, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# role abort` | Discards the role-based configuration changes and clears the pending configuration database. |

# Enabling Role-Based Configuration Distribution

To enable role-based configuration distribution, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# role distribute` | Enables role-based configuration distribution. |
| | `switch(config)# no role distribute` | Disables role-based configuration distribution (default). |

# Clearing Sessions

To forcibly clear the existing role session in the fabric, issue the **clear role session** command from any switch that is part of the initiated session.

**Caution** Any changes in the pending database are lost when you issue this command.

```
switch# clear role session
```

# Database Merge Guidelines

Fabric merge does not modify the role database on a switch. If two fabrics merge, and the fabrics have different role databases, the software generates an alert message.

- Verify that the role database is identical on all switches in the entire fabric.
- Be sure to edit the role database on any switch to the desired database and then commit it. This synchronizes the role databases on all the switches in the fabric.

# Displaying Role-Based Information

Use the **show role** command to display rules configured on the switch. The rules are displayed by rule number and are based on each role. All roles are displayed if the role name is not specified. See Example 3-1.

*Example 3-1 Displays Information for All Roles*

```
switch# show role
Role: network-admin
  Description: Predefined Network Admin group. This role cannot be modified.
  Vsan policy: permit (default)
  ----------------------------------------------
  Rule    Type    Command-type    Feature
  ----------------------------------------------
  1       permit  clear           *
  2       permit  config          *
  3       permit  debug           *
  4       permit  exec            *
  5       permit  show            *

Role: network-operator
  Description: Predefined Network Operator group. This role cannot be modified.
  Vsan policy: permit (default)
  ----------------------------------------------
  Rule    Type    Command-type    Feature
  ----------------------------------------------
  1       permit  show            *(excluding show running-config, show startup-config)
  2       permit  exec            copy licenses
  3       permit  exec            dir
  4       permit  exec            ssh
  5       permit  exec            terminal
  6       permit  config          username

Role: server-admin
  Description: Predefined system role for server administrators. This role
  cannot be modified.
  Vsan policy: permit (default)
  ----------------------------------------------
  Rule    Type    Command-type    Feature
  ----------------------------------------------
  1       permit  show            *
  2       permit  exec            install

Role: priv-15
  Description: This is a system defined privilege role.
  Vsan policy: permit (default)
  ----------------------------------------------
  Rule    Type    Command-type    Feature
  ----------------------------------------------
  1       permit  show            *
  2       permit  config          *
  3       permit  clear           *
  4       permit  debug           *
  5       permit  exec            *

Role: priv-14
  Description: This is a system defined privilege role.
  Vsan policy: permit (default)

Role: priv-13
  Description: This is a system defined privilege role.
  Vsan policy: permit (default)

Role: priv-12
  Description: This is a system defined privilege role.
  Vsan policy: permit (default)

Role: priv-11
  Description: This is a system defined privilege role.
```

```
  Vsan policy: permit (default)

Role: priv-10
  Description: This is a system defined privilege role.
  Vsan policy: permit (default)

Role: priv-9
  Description: This is a system defined privilege role.
  Vsan policy: permit (default)

Role: priv-8
  Description: This is a system defined privilege role.
  Vsan policy: permit (default)

Role: priv-7
  Description: This is a system defined privilege role.
  Vsan policy: permit (default)

Role: priv-6
  Description: This is a system defined privilege role.
  Vsan policy: permit (default)

Role: priv-5
  Description: This is a system defined privilege role.
  Vsan policy: permit (default)

Role: priv-4
  Description: This is a system defined privilege role.
  Vsan policy: permit (default)

Role: priv-3
  Description: This is a system defined privilege role.
  Vsan policy: permit (default)

Role: priv-2
  Description: This is a system defined privilege role.
  Vsan policy: permit (default)

Role: priv-1
  Description: This is a system defined privilege role.
  Vsan policy: permit (default)

Role: priv-0
  Description: This is a system defined privilege role.
  Vsan policy: permit (default)
  ----------------------------------------------------
  Rule    Type    Command-type    Feature
  ----------------------------------------------------
  1       permit  show            *
  2       permit  exec            enable
  3       permit  exec            ssh
  4       permit  exec            ping
  5       permit  exec            telnet
  6       permit  exec            traceroute

Role: default-role
  Description: This is a system defined role and applies to all users.
  Vsan policy: permit (default)
  ----------------------------------------------------
  Rule    Type    Command-type    Feature
  ----------------------------------------------------
  1       permit  show            system
  2       permit  show            snmp
  3       permit  show            module
```

```
  4       permit  show            hardware
  5       permit  show            environment
```

# Displaying Roles When Distribution is Enabled

Use the **show role** command to display the configuration database.

Use the **show role status** command to display whether distribution is enabled for role configuration, the current fabric status (locked or unlocked), and the last operation performed. See Example 3-2.

***Example 3-2 Displays the Role Status Information***

```
switch# show role status
Distribution: Enabled
Session State: Locked

Last operation (initiated from this switch): Distribution enable
Last operation status: Success
```

Use the **show role pending** command to display the pending role database.

Example 3-3 displays the output of the **show role pending** command by following this procedure:

1. Create the role called `myrole` using the **role name myrole** command.
2. Enter the **rule 1 permit config feature fspf** command.
3. Enter the **show role pending** command to see the output.

***Example 3-3 Displays Information on the Pending Roles Database***

```
switch# show role pending
Role: network-admin
Description: Predefined Network Admin group. This role cannot be modified
Access to all the switch commands

Role: network-operator
Description: Predefined Network Operator group. This role cannot be modified
Access to Show commands and selected Exec commands

Role: svc-admin
Description: Predefined SVC Admin group. This role cannot be modified
Access to all SAN Volume Controller commands

Role: svc-operator
Description: Predefined SVC Operator group. This role cannot be modified
Access to selected SAN Volume Controller commands

Role: TechDocs
  vsan policy: permit (default)

Role: sangroup
  Description: SAN management group
  vsan policy: deny
  Permitted vsans: 10-30

  ----------------------------------------------
  Rule    Type    Command-type      Feature
  ----------------------------------------------
    1.   permit    config                   *
```

```
    2.     deny    config                 fspf
    3.   permit     debug                 zone
    4.   permit      exec               fcping

Role: myrole
  vsan policy: permit (default)
  ----------------------------------------------
  Rule    Type    Command-type      Feature
  ----------------------------------------------
    1.   permit    config                 fspf
```

Use the **show role pending-diff** command to display the differences between the pending and configuration role database. See Example 3-4.

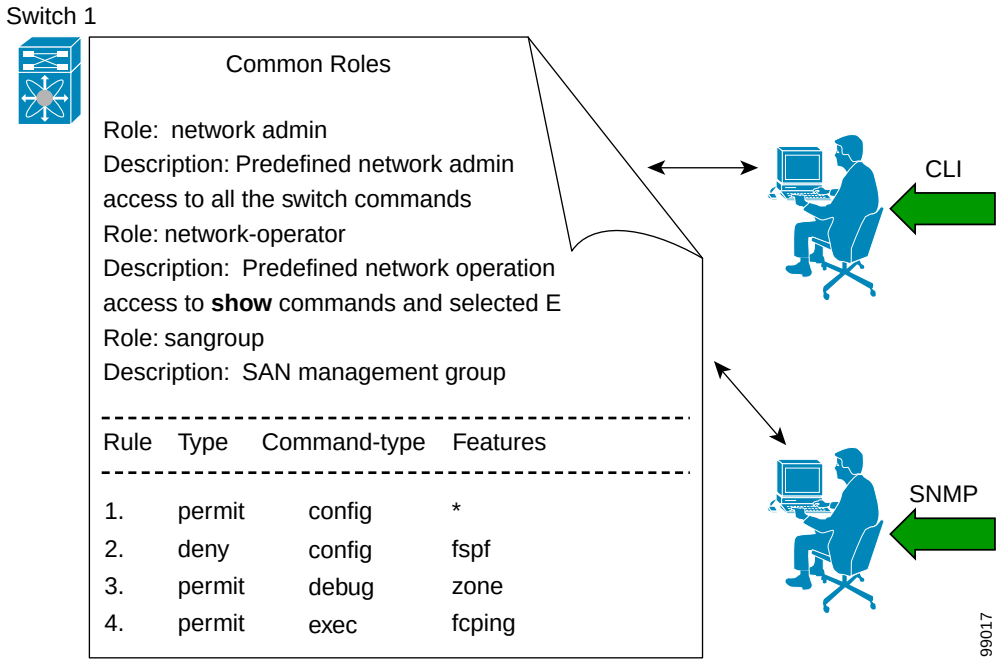
***Example 3-4 Displays the Differences Between the Two Databases***

```
switch# show role pending-diff
    +Role: myrole
    +  vsan policy: permit (default)
    +  ----------------------------------------------
    +  Rule    Type    Command-type      Feature
    +  ----------------------------------------------
    +    1.   permit    config                fspf
```

# Configuring Common Roles

The CLI and SNMP in all switches in the Cisco MDS 9000 Family use common roles. You can use SNMP to modify a role that was created using the CLI and vice versa (see Figure 3-1).

***Figure 3-1 Common Roles***



A custom role user with Network-Admin privileges is restricted to modify the account of other users. However, only the Admin can modify all user accounts.

You can modify the user privileges by performing the following task.

1. Modify role using console authentication.

If you setup the console authentication as 'local', logon using the Local-Admin user and modify the user.

2. Modify role using remote authentication.

Turn off the remote authentication. Logon using the Local -Admin privileges and modify the user. Turn on the remote authentication.

3. Modify role using LDAP/AAA.

Create a group in LDAP/AAA and rename the group as Network-Admin. Add the required users to this group. The users of this group will now have complete Network-Admin privileges.

Each role in SNMP is the same as a role created or modified through the CLI (see the "Role-Based Authorization" section on page 3-32).

Each role can be restricted to one or more VSANs as required.

You can create new roles or modify existing roles using SNMP or the CLI.

- SNMP—Use the CISCO-COMMON-ROLES-MIB to configure or modify roles. Refer to the *Cisco MDS 9000 Family MIB Quick Reference*.
- CLI—Use the **role name** command.

# Mapping of CLI Operations to SNMP

SNMP has only three possible operations: GET, SET, and NOTIFY. The CLI has five possible operations: DEBUG, SHOW, CONFIG, CLEAR, and EXEC.

**Note** NOTIFY does not have any restrictions like the syslog messages in the CLI.

Table 3-2 explains how the CLI operations are mapped to the SNMP operations.

*Table 3-2 CLI Operation to SNMP Operation Mapping*

| CLI Operation | SNMP Operation |
|---|---|
| DEBUG | Ignored |
| SHOW | GET |
| CONFIG | SET |
| CLEAR | SET |
| EXEC | SET |

Example 3-5 shows the privileges and rules mapping CLI operations to SNMP operations for a role named my_role.

*Example 3-5 Displays CLI Operation to SNMP Operation Mapping*

```
switch# show role name my_role
Role:my_role
  vsan policy:permit (default)
  ----------------------------------------------
  Rule    Type    Command-type      Feature
  ----------------------------------------------
    1.   permit     clear                 *
    2.     deny     clear               ntp
    3.   permit    config                 *
    4.     deny    config               ntp
    5.   permit     debug                 *
    6.     deny     debug               ntp
    7.   permit      show                 *
    8.     deny      show               ntp
    9.   permit      exec                 *
```

**Note** Although CONFIG is denied for NTP in rule 4, rule 9 allows the SET to NTP MIB objects because EXEC also maps to the SNMP SET operation.

# Configuring User Accounts

Every Cisco MDS 9000 Family switch user has the account information stored by the system. Your authentication information, user name, user password, password expiration date, and role membership are stored in your user profile.

The tasks explained in this section enable you to create users and modify the profile of an existing user. These tasks are restricted to privileged users as determined by your administrator.

This section includes the following topics:



## Creating Users Guidelines

The passphrase specified in the **snmp-server user** option and the password specified **username** option are synchronized.

By default, the user account does not expire unless you explicitly configure it to expire. The **expire** option determines the date on which the user account is disabled. The date is specified in the YYYY-MM-DD format.

When creating users, note the following guidelines:

- You can configure up to a maximum of 256 users on a switch.
- The following words are reserved and cannot be used to configure users: bin, daemon, adm, lp, sync, shutdown, halt, mail, news, uucp, operator, games, gopher, ftp, nobody, nscd, mailnull, rpc, rpcuser, xfs, gdm, mtsuser, ftpuser, man, and sys.
- User passwords are not displayed in the switch configuration file.

- If a password is trivial (short, easy-to-decipher), your password configuration is rejected. Be sure to configure a strong password as shown in the sample configuration. Passwords are case-sensitive. “admin” is no longer the default password for any Cisco MDS 9000 Family switch. You must explicitly configure a strong password.
- To issue commands with the **internal** keyword for troubleshooting purposes, you must have an account that is a member of the network-admin group.

**Caution** Cisco MDS NX-OS supports user names that are created with alphanumeric characters or specific special characters (+ [plus], = [equal], _ [underscore], - [hyphen], \ [backslash], and . [period]) whether created remotely (using TACACS+ or RADIUS) or locally, provided that the user name starts with an alphanumeric character. Local user names cannot be created with any special characters (apart from those specified). If a nonsupported special character user name exists on an AAA server, and is entered during login, then the user is denied access.

## Checking Password Strength

You can check the strength of the configured password.

When you enable password checking, the NX-OS software allows you to create strong passwords only.

To enable password strength checking, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# password strength-check` | Enables (default) password checking. |
| **Step 3** | `switch(config)# no password strength-check` | Disables password checking. |

## Characteristics of Strong Passwords

A strong password has the following characteristics:

- At least eight characters long
- Does not contain many consecutive characters (such as “abcd”)
- Does not contain many repeating characters (such as “aaabbb”)
- Does not contain dictionary words
- Does not contain proper names
- Contains both upper- and lower-case characters
- Contains numbers

The following are examples of strong passwords:

- If2CoM18
- 2004AsdfLkj30
- Cb1955S21

# Configuring Users

To configure a new user or to modify the profile of an existing user, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# username usam password abcd123AAA expire 2003-05-31` | Creates or updates the user account (usam) along with a password (abcd123AAA) that is set to expire on 2003-05-31. |
| | `switch(config)# username msam password 0 abcd12AAA role network-operator` | Creates or updates the user account (msam) along with a password (abcd12AAA) specified in clear text (indicated by 0). The password is limited to 64 characters. |
| | `switch(config)# username user1 password 5 $1$UgOR6Xqb$z.HZlMk.ZGr9VH67a` | Specifies an encrypted (specified by 5) password (!@*asdsfsdfjh!@df) for the user account (user1).<br>**Note** If user is created with encrypted password option then corresponding SNMP user will not be created. |
| **Step 3** | `switch(config)# username usam role network-admin` | Adds the specified user (usam) to the network-admin role. |
| | `switch(config)# no username usam role vsan-admin` | Deletes the specified user (usam) from the vsan-admin role. |
| **Step 4** | `switch(config)# username admin sshkey ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAtjIHrIt/3dDeohix6JcRSI YZ0EOdJ3l5RONWcwSgAuTUSrLk 3a9hdYkzY94fhHmNGQGCjVg+8cbOxyH4Z1jcVFcrDogtQT+Q8d veqts/8XQhqkNAFeGy4u8TJ2Us oreCU6DlibwkpzDafzKTpA5vB6FmHd2TI6Gnse9FUgKD5fs=` | Specifies the SSH key for an existing user account (admin). |
| | `switch(config)# no username admin sshkey ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAtjIHrIt/3dDeohix6JcRSI YZ0EOdJ3l5RONWcwSgAuTUSrLk 3a9hdYkzY94fhHmNGQGCjVg+8cbOxyH4Z1jcVFcrDogtQT+Q8d veqts/8XQhqkNAFeGy4u8TJ2Us oreCU6DlibwkpzDafzKTpA5vB6FmHd2TI6Gnse9FUgKD5fs=` | Deletes the SSH key for the user account (admin). |
| **Step 5** | `switch(config)# username usam ssh-cert-dn usam-dn dsa` | Specifies an SSH X.509 certificate distinguished name and DSA algorithm to use for authentication for an existing user account (usam). |
| | `switch(config)# username user1 ssh-cert-dn user1-dn rsa` | Specifies an SSH X.509 certificate distinguished name and RSA algorithm to use for authentication for an existing user account (user1). |
| | `switch(config)# no username admin ssh-cert-dn admin-dn dsa` | Removes the SSH X.509 certificate distinguished name for the user account (admin). |

# Logging Out Users

To log out another user on the switch, use the **clear user** command.

In the following example, the user named vsam is logged out from the switch:

```
switch# clear user vsam
```

Use the **show users** command to view a list of the logged in users (see Example 3-6).

***Example 3-6 Displays All Logged in Users***

```
switch# show users
admin    pts/7        Jan 12 20:56 (10.77.202.149)
admin    pts/9        Jan 12 23:29 (user.example.com)
admin    pts/10       Jan 13 03:05 (dhcp-10-10-1-1.example.com)
admin    pts/11       Jan 13 01:53 (dhcp-10-10-2-2.example.com)
```

# Displaying User Account Information

Use the **show user-account** command to display configured information about user accounts. See Examples 3-7 to 3-8.

***Example 3-7 Displays Information for a Specified User***

```
switch# show user-account user1
user:user1
        this user account has no expiry date
        roles:network-operator
no password set. Local login not allowed
Remote login through RADIUS is possible
```

***Example 3-8 Displays Information for All Users***

```
switch# show user-account
show user-account
user:admin
        this user account has no expiry date
        roles:network-admin
user:usam
        expires on Sat May 31 00:00:00 2003
        roles:network-admin network-operator
user:msam
        this user account has no expiry date
        roles:network-operator
user:user1
        this user account has no expiry date
        roles:network-operator
no password set. local login not allowed
Remote login through RADIUS is possible
```

# Secure Login Enhancements

The following secure login enhancements are supported in Cisco MDS 9000 Series Switches:



## Configuring Login Parameters

Use this task to configure your Cisco MDS 9000 device for login parameters that helps to detect suspected DoS attacks and slow down dictionary attacks.

All login parameters are disabled by default. You must enter the **login block-for** command, which enables default login functionality, before using any other login commands. After the **login block-for** command is enabled, the following default is enforced:

- All login attempts made through Telnet or SSH are denied during the quiet period; that is, no ACLs are exempt from the login period until the **login quiet-mode access-class** command is entered.

To configure the login parameter, follow these steps:

**Step 1** Enter the global configuration mode:

```
switch# configure terminal
```

**Step 2** Configure your Cisco MDS 9000 device for login parameters that helps to provide DoS detection:

```
switch(config)# system login block-for seconds attempts tries within seconds
```

**Note** This command must be issued before any other login command.

**Step 3** (Optional) Although this command is optional, it is recommended that, it should be configured to specify an ACL that is to be applied to the device when the device switches to quiet mode. When the device is in quiet mode, all login requests are denied and the only available connection is through the console:

```
switch(config)# system login quiet-mode access-class {acl-name | acl-number}
```

**Step 4** Exit to privileged EXEC mode:

```
switch(config)# exit
```

**Step 5** Display login parameters:

```
switch# show system login
```

**Step 6** Display information related only to failed login attempts:

```
switch# show system login failures
```

*Example 3-9 Setting Login Parameters*

The following example shows how to configure your switch to enter into a 100 seconds quiet period if 15 failed login attempts is exceeded within 100 seconds. All login requests are denied during the quiet period except hosts from the ACL "myacl."

```
switch(config)# system login block-for 100 attempts 15 within 100
switch(config)# system login quiet-mode access-class myacl
```

*Example 3-10 Displays default ACLs*

The following sample output from the **show ip access-list sl_def_acl** command displays default ACLs.

```
switch(config)# show ip access-list sl_def_acl
ip access-list sl_def_acl
permit tcp any any established (0 matches)
deny tcp any any eq port telnet (0 matches)
deny tcp any any eq port www (0 matches)
deny tcp any any eq port ssh (0 matches)
permit ip any any (0 matches)
```

*Example 3-11 Verifies no login parameters*

The following sample output from the **show system login** command verifies that no login parameters have been specified.

```
switch# show system login
No Quiet-Mode access list has been configured, default ACL will be applied.

Switch is enabled to watch for login Attacks.
 If more than 2 login failures occur in 20 seconds or less,
logins will be disabled for 60 seconds.

Switch presently in Quiet-Mode.

Will remain in Quiet-Mode for 43 seconds.

Denying logins from all sources.
```

*Example 3-12 Displays information on failed login attempts*

The following sample output from the **show system login failures** command shows all failed login attempts on the switch:

```
switch# show system login failures
Information about last 20 login failure's with the device.
-------------------------------------------------------------------------------------------
---------------
-
Username                                Line    Source          Appname         TimeStamp
-------------------------------------------------------------------------------------------
---------------
-
-------------------------------------------------------------------------------------------
---------------
-
lock4                                   pts/1   192.0.2.2  login           Thu Feb 16
14:36:12 2017
as                                      pts/1   192.0.2.2   login           Thu Feb 16
14:36:16 2017
as                                      pts/1   192.0.2.2  login           Thu Feb 16
14:36:20 2017
```

# Configuring Login Block Per User

The Login Block Per User feature helps detect suspected Denial of Service (DoS) attacks and to slow down dictionary attacks. This feature is applicable only for local users. Use this task to configure login parameters to block an user after failed login attempts.

To configure login block per user, follow these steps:

**Step 1** Enter the global configuration mode:

```
switch# configure terminal
```

**Step 2** Configure login parameters to block a user:

```
switch(config)# aaa authentication rejected attempts in seconds ban seconds
```

**Note** Use the **no aaa authentication rejected** command to revert to the default login parameters.

**Step 3** Exit to privileged EXEC mode:

```
switch(config)# exit
```

**Step 4** Display login parameters:

```
switch# show system login
```

**Step 5** Display the blocked local users:

```
switch# show aaa local user blocked
```

**Step 6** Clear blocked local users.:

```
switch# clear aaa local user blocked {username user | all}
```

*Example 3-13 Configuring login block per user*

The following example shows how to configure the login parameters to block a user for 300 seconds when five login attempts fail within a period of 60 seconds:

```
switch# aaa authentication rejected 5 in 60 ban 3
```

*Example 3-14 Displays login parameters*

The following example shows the login parameters configured for a switch:

```
switch# show run | i rejected
aaa authentication rejected 5 in 60 ban 300
```

*Example 3-15 Displays blocked local users*

The following example shows the blocked local users:

```
switch# show aaa local user blocked
Local-user                  State

testuser                    Watched (till 11:34:42 IST Feb 5 2015)
```

*Example 3-16 Clears blocked local users*

The following example shows how to clear the blocked local user testuser:

```
switch# clear aaa local user blocked username testuser
```

# Restricting Sessions Per User—Per User Per Login

To restrict the maximum sessions per user, follow these steps:

**Step 1** Enter the global configuration mode:

```
switch# configure terminal
```

**Step 2** Restrict the maximum sessions per user. The range is from 1 to 7. If you set the maximum login limit as 1, then only one session (telnet/SSH) is allowed per user:

```
switch(config)# user max-logins max-logins
```

**Step 3** Exit to privileged EXEC mode:

```
switch(config)# exit
```

*Example 3-17 Restricting sessions per user*

The following example shows how to restrict the maximum number of logins per user to 1 session:

```
switch# user max-logins 1
```

# Configuring Passphrase and Locking User Accounts

To configure passphrase lengths, time values, and locking user accounts, follow these steps:

**Step 1** Enter the global configuration mode:

```
switch# configure terminal
```

**Step 2** Configure either the minimum or maximum passphrase length.

```
switch(config)# userpassphrase {min-length min_value | max-length max_value}
```

**Step 3** Display the minimum, maximum, or complete passphrase length configuration:

```
switch# show userpassphrase {min-length | max-length | length}
```

**Step 4** Configure passphrase lifetimes for any user:

```
switch(config)# username user passphrase {lifetime | warntime | gracetime}
```

**Step 5** (Optional) Update default configurations:

```
switch(config)# userpassphrase {default-lifetime | default-warntime | default-gracetime |
min-length min_value | max-length max_value}
```

**Step 6** Display passphrase lifetimes configured for any user:

```
switch# show username user passphrase timevalues
```

**Step 7** Lock any user account:

```
switch(config)# username user lock-user-account
```

**Step 8** Expire any userpassphrase:

```
switch(config)# username user expire-userpassphrase
```

**Step 9** Display all locked users:

```
switch(config)# show locked-users
```

*Example 3-18 Configuring Maximum and Minimum Passphrase Lengths*

The following example shows how to configure the minimum passphrase length as 8 and maximum passphrase length as 80:

```
switch(config)# userpassphrase min-length 8 max-length 80
```

*Example 3-19 Displays Minimum Passphrase Length*

The following example shows the minimum passphrase length:

```
switch(config)# show userpassphrase min-length
Minimum passphrase length : 8
```

*Example 3-20 Configuring Passphrase Lifetime Values for a User*

The following example shows how to configure the passphrase lifetime values for a user:

```
switch(config)# username user1 passphrase lifetime 10
```

*Example 3-21 Displays Passphrase Lifetime Values for a User*

The following example shows how to configure the passphrase lifetime values for a user:

```
switch(config)# show username user1 passphrase timevalues
Last passphrase change(Y-M-D):  2017-02-06
Passphrase lifetime:            99999 days after last passphrase change
Passphrase warning time starts: 7 days before passphrase lifetime
Passphrase Gracetime ends:      never
```

*Example 3-22 Locking a User Account*

The following example shows how to lock a user account:

```
switch(config)# username user1 lock-user-account
```

*Example 3-23 Expiring a Userpassphrase*

The following example shows how to lock a user account:

```
switch(config)# username user1 expire-userpassphrase
```

*Example 3-24 Displays Locked Users*

The following example shows all locked users:

```
switch(config)# show locked-users
```

# Enabling the Password Prompt for User Name

To enable the password prompt for a user name, follow these steps:

**Step 1** Enter the global configuration mode:

```
switch# configure terminal
```

**Step 2** Enable the login knob. If this command is enabled and the user enters the **username** command without the *password* option, then the password is prompted. The password accepts hidden characters. Use the **no** form of this command to disable the login knob.:

```
switch(config)# password prompt username
```

**Step 3** Exit to privileged EXEC mode:

```
switch(config)# exit
```

# Support over SHA-256 Algorithm for Verifying OS Integrity

Use the **show file bootflash:/ sha256sum** command to display the sha256sum of the file. The sample output for this command is shown below:

```
switch# show file bootflash:/ sha256sum

abd9d40020538acc363df3d1bae7d1df16841e4903fca2c07c7898bf4f549ef5
```

# Configuring Share Key Value for using RADIUS/TACACS+

The shared secret you configure for remote authentication and accounting must be hidden. For the **radius-server key** and **tacacs-server key** commands, a separate command to generate encrypted shared secret can be used.

To configure the share key value for using RADIUS/TACACS+, follow these steps:

**Step 1** Enter the global configuration mode:

```
switch# configure terminal
```

**Step 2** Configure RADIUS and TACACS shared secret with key type 7. While generating an encrypted shared secret, user input is hidden:

```
switch(config)# generate type7_encrypted_secret
```

**Note** You can generate encrypted equivalent of plain text separately and can configure the encrypted shared secret later.

**Step 3** Exit to privileged EXEC mode:

```
switch(config)# exit
```

# Configuring SSH

A secure SSH connection, with rsa key is available as default on all Cisco MDS 9000 Family switches. If you require a secure SSH connection with dsa key, you need to disable the default SSH connection, Generate a dsa key and then enable the SSH connection (see the "Generating the SSH Server Key Pair" section on page 3-55).

Use the **ssh key** command to generate a server key.

**Caution** If you are logging in to a switch through SSH and you have issued the **aaa authentication login default none** command, you must enter one or more key strokes to log in. If you press the **Enter** key without entering at least one keystroke, your log in will be rejected.

This section includes the following topics:



# About SSH

SSH provides secure communications to the Cisco NX-OS CLI. You can use SSH keys for the following SSH options:

- SSH2 using RSA
- SSH2 using DSA

# Generating the SSH Server Key Pair

Be sure to have an SSH server key pair with the appropriate version before enabling the SSH service. Generate the SSH server key pair according to the SSH client version used. The number of bits specified for each key pair ranges from 768 to 2048.

The SSH service accepts two types of key pairs for use by SSH version 2.

- The **dsa** option generates the DSA key pair for the SSH version 2 protocol.
- The **rsa** option generates the RSA keypair for the SSH version 2 protocol.

**Caution** If you delete all of the SSH keys, you cannot start a new SSH session.

To generate the SSH server key pair, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |

| | Command | Purpose |
|---|---|---|
| Step 2 | `switch(config)# ssh key dsa 1024`<br>`generating dsa key.....`<br>`generated dsa key` | Generates the DSA server key pair. |
| | `switch(config)# ssh key rsa 1024`<br>`generating rsa key.....`<br>`generated rsa key` | Generates the RSA server key pair. |
| | `switch(config)# no ssh key rsa 1024`<br>`cleared RSA keys` | Clears the RSA server key pair configuration. |

# Specifying the SSH Key

You can specify an SSH key to log in using the SSH client without being prompted for a password. You can specify the SSH key in three different formats:

- Open SSH format
- IETF SECSH format
- Public Key Certificate in PEM format

To specify or delete the SSH key in OpenSSH format for a specified user, follow these steps:

| | Command | Purpose |
|---|---|---|
| Step 1 | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| Step 2 | `switch(config)# username admin sshkey ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAtjIHrIt/3dDeohix6JcRSIYZ 0EOdJ3l5RONWcwSgAuTUSrLk3a9hdYkzY94fhHmNGQGCjVg+8cbO xyH4Z1jcVFcrDogtQT+Q8dveqts/8XQhqkNAFeGy4u8TJ2UsoreC U6DlibwkpzDafzKTpA5vB6FmHd2TI6Gnse9FUgKD5fs=` | Specifies the SSH key for the user account (admin). |
| | `switch(config)# no username admin sshkey ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAtjIHrIt/3dDeohix6JcRSIYZ 0EOdJ3l5RONWcwSgAuTUSrLk3a9hdYkzY94fhHmNGQGCjVg+8cbO xyH4Z1jcVFcrDogtQT+Q8dveqts/8XQhqkNAFeGy4u8TJ2UsoreC U6DlibwkpzDafzKTpA5vB6FmHd2TI6Gnse9FUgKD5fs=` | Deletes the SSH key for the user account (admin). |

To specify or delete the SSH key in IETF SECSH format for a specified user, follow these steps:

| | Command | Purpose |
|---|---|---|
| Step 1 | `switch# copy tftp://10.10.1.1/secsh_file.pub bootflash:secsh_file.pub` | Downloads the file containing the SSH key in IETF SECSH format. |
| Step 2 | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| Step 3 | `switch(config)# username admin sshkey file bootflash:secsh_file.pub` | Specifies the SSH key for the user account (admin). |
| | `switch(config)# no username admin sshkey file bootflash:secsh_file.pub` | Deletes the SSH key for the user account (admin). |

To specify or delete the SSH key in PEM-formatted Public Key Certificate form for a specified user, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# copy tftp://10.10.1.1/cert.pem bootflash:cert.pem` | Downloads the file containing the SSH key in PEM-formatted Public Key Certificate form. |
| **Step 2** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 3** | `switch(config)# username admin sshkey file bootflash:cert.pem` | Specifies the SSH key for the user account (usam). |
| | `switch(config)# no username admin sshkey file bootflash:cert.pem` | Deletes the SSH key for the user account (usam). |

# Overwriting a Generated Key Pair

If the SSH key pair option is already generated for the required version, you can force the switch to overwrite the previously generated key pair.

To overwrite the previously generated key pair, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# ssh key dsa 768`<br>`ssh key dsa 512`<br>`dsa keys already present, use force`<br>`option to overwrite them`<br>`switch(config)# ssh key dsa 512 force`<br>`deleting old dsa key.....`<br>`generating dsa key.....`<br>`generated dsa key` | Tries to set the server key pair. If a required server key pair is already configured, use the **force** option to overwrite that server key pair.<br>Deletes the old DSA key and sets the server key pair using the new bit specification. |

# Clearing SSH Hosts

The **clear ssh hosts** command clears the existing list of trusted SSH hosts and reallows you to use SCP/SFTP along with the **copy** command for particular hosts.

When you use SCP/SFTP along with the **copy** command, a list of trusted SSH hosts are built and stored within the switch (see Example 3-25).

***Example 3-25 Using SCP/SFTP to Copy Files***

```
switch# copy scp://abcd@10.10.1.1/users/abcd/abc
bootflash:abc The authenticity of host '10.10.1.1 (10.10.1.1)'
can't be established.
RSA1 key fingerprint is 01:29:62:16:33:ff:f7:dc:cc:af:aa:20:f8:20:a2:db.
Are you sure you want to continue connecting (yes/no)? yes
Added the host to the list of known hosts
(/var/home/admin/.ssh/known_hosts). [SSH key information about the host is
stored on the switch]
abcd@10.10.1.1's password:
switch#
```

If a host's SSH key changes before you use SCP/SFTP along with the **copy** command, you will receive an error (see Example 3-26).

***Example 3-26 Using SCP/SFTP to Copy Files—Error Caused by SSH Key Change***

```
switch# copy scp://apn@10.10.1.1/isan-104
bootflash:isan-ram-1.0.4
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA1 host key has just been changed.
The fingerprint for the RSA1 key sent by the remote host is
36:96:ca:d7:29:99:79:74:aa:4d:97:49:81:fb:23:2f.
Please contact your system administrator.
Add correct host key in /mnt/pss/.ssh/known_hosts to get rid of this
message.
Offending key in /mnt/pss/.ssh/known_hosts:2
RSA1 host key for 10.10.1.1 has changed and you have requested strict
checking.
```

# Enabling SSH or Telnet Service

By default, the SSH service is enabled with the rsa key.

To enable or disable the SSH or Telnet service, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# feature ssh`<br>`updated` | Enables the use of the SSH service. |
| | `switch(config)# no feature ssh`<br>`updated` | Disables (default) the use of the SSH service. |
| | `switch(config)# feature telnet`<br>`updated` | Enables the use of the Telnet service. |
| | `switch(config)# no feature telnet`<br>`updated` | Disables (default) the use of the Telnet service. |

# Displaying SSH Protocol Status

Use the **show ssh server** command to display the status of the SSH protocol (enabled or disabled) and the versions that are enabled for that switch (see Example 3-27).

***Example 3-27 Displays SSH Protocol Status***

```
switch# show ssh server
ssh is enabled
version 1 enabled
version 2 enabled
```

Use the **show ssh key** command to display the server key-pair details for the specified key or for all keys, (see Example 3-28).

*Example 3-28 Displays Server Key-Pair Details*

```
switch# show ssh key
rsa1 Keys generated:Sun Jan 13 07:16:26 1980
1024 35
fingerprint:
1024 67:76:02:bd:3e:8d:f5:ad:59:5a:1e:c4:5e:44:03:07
could not retrieve rsa key information
dsa Keys generated:Sun Jan 13 07:40:08 1980
ssh-dss
AAAAB3NzaC1kc3MAAABBAJTCRQOydNRel2v7uiO6Fix+OTn8eGdnnDVxw5eJs5OcOEXOyjaWcMMYsEgxc9ada1NElp
8Wy7GPMWGOQYj9CU0AAAAVAMCcWhNN18zFNOIPo7cU3t7d0iEbAAAAQBdQ8UAOi/Cti84qFb3kTqXlS9mEhdQUo0lH
cH5bw5PKfj2Y/dLR437zCBKXetPj4p7mhQ6Fq5os8RZtJEyOsNsAAABAA0oxZbPyWeR5NHATXiyXdPI7j9i8fgyn9F
NipMkOF2Mn75Mi/lqQ4NIq0gQNvQOx27uCeQlRts/QwI4q68/eaw=
fingerprint:
512 f7:cc:90:3d:f5:8a:a9:ca:48:76:9f:f8:6e:71:d4:ae
```

**Note** If you are logging in to a switch through SSH and you have issued the **aaa authentication login default none** CLI command, you must enter one or more key strokes to log in. If you press the **Enter** key without entering at least one keystroke, your log in will be rejected.

# SSH Authentication Using Digital Certificates

SSH authentication on the Cisco MDS 9000 Family switches provide X.509 digital certificate support for host authentication. An X.509 digital certificate is a data item that vouches for the origin and integrity of a message. It contains encryption keys for secured communications and is “signed” by a trusted certification authority (CA) to verify the identity of the presenter. The X.509 digital certificate support provides either DSA or RSA algorithms for authentication.

The certificate infrastructure uses the first certificate that supports the Secure Socket Layer (SSL) and is returned by the security infrastructure, either through query or notification. Verification of certificates is successful if the certificates are from any of the trusted CAs.

You can configure your switch for either SSH authentication using an X.509 certificate or SSH authentication using a Public Key Certificate, but not both. If either of them is configured and the authentication fails, you will be prompted for a password.

# Passwordless File copy and SSH

Secure Shell (SSH) public key authentication can be used to achieve password free logins. SCP and SFTP uses SSH in the background and hence these copy protocols can be used for a password free copy with public key authentication. The NX-OS version only supports the SCP and STFP client functionality.

You can create an RSA/DSA identity which can be used for authentication with ssh. The identity will consist of two parts: public and private keys. The public and the private keys are generated by the switch or can be generated externally and imported to the switch. For import purposes, the keys should be in OPENSSH format.

To use the key on a host machine hosting an SSH server, you must transfer the public key file to the machine and add the contents of it to the file 'authorized_keys' in your ssh directory (e.g. $HOME/.ssh) on the server. For import and export of private keys, the key will be protected by encryption. You will be asked to enter a Passphrase for the same. If you enter a passphrase, the private key is protected by encryption. If you leave the password field blank, the key will not be encrypted.

If you need to copy the keys to another switch, you will have to export the keys out of the switch to a host machine and then import the same to other switches from that machine.

- The key files are persistent across reload.

To import and export the key pair, the following CLIs are provided. The CLI command to generate the ssh user key pairs on the switch is defined as follows:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# username admin keypair generate rsa`<br>`generating rsa key(1024 bits).....`<br>`generated rsa key` | Generates public and private RSA keys for the account (admin). It then stores the key files in the home directory of the specified user. Use the force option to overwrite that server keypair.<br>**Note** This example is for RSA keys. Replace rsa with dsa for DSA keys. |
| | `switch(config)# no username admin keypair generate rsa` | Deletes the public and private RSA keys for the account (admin). |

| | Command | Purpose |
|---|---|---|
| **Step 3** | `switch# show username admin keypair`<br>`**********`<br>`rsa Keys generated: Thu Jul 9 11:10:29 2009`<br>`ssh-rsa`<br>`AAAAB3NzaC1yc2EAAAABIwAAAIEAxWmjJT+oQhIcvnrMbx2BmD`<br>`0P8boZElTfJFx9fexWp6rOiztlwODtehnjadWc6A+DE2DvYNvq`<br>`srU9TBypYDPQkR/+Y6cKubyFWVxSBG/NHztQc3+QC1zdkIxGNJ`<br>`bEHyFoajzNEO8LLOVFIMCZ2Td7gxUGRZc+fbqS33GZsCAX6v0=`<br>`bitcount:262144`<br>`fingerprint:`<br>`8d:44:ee:6c:ca:0b:44:95:36:d0:7d:f2:b5:78:74:7d`<br>`**********`<br>`could not retrieve dsa key information`<br>`**********` | Shows the public key for the account (admin). |
| **Step 4** | `switch(config)# username admin keypair export bootflash:key_rsa rsa`<br>`Enter Passphrase:`<br>`switch(config)# dir`<br>`951 Jul 09 11:13:59 2009 key_rsa`<br>`221 Jul 09 11:14:00 2009 key_rsa.pub` | Exports the keypair from the user's (admin's) home directory to the bootflash memory.<br>The key pair (both public and private keys) will be exported to the specified location. The user will be prompted to enter a Passphrase which will encrypt the private key. The private key will be exported as the file name specified in the uri and the public key will be exported with the same file name followed by a ".pub" extension.<br>The user can now copy this key pair to any switch, and also copy the public file to the home directory of the SCP server. |
| **Step 5** | `switch(config)# username admin keypair import bootflash:key_rsa rsa`<br>`Enter Passphrase:`<br>`switch(config)# show username admin keypair`<br>`**********`<br>`rsa Keys generated: Thu Jul 9 11:10:29 2009`<br>`ssh-rsa`<br>`AAAAB3NzaC1yc2EAAAABIwAAAIEAxWmjJT+oQhIcvnrMbx2BmD`<br>`0P8boZElTfJFx9fexWp6rOiztlwODtehnjadWc6A+DE2DvYNvq`<br>`srU9TBypYDPQkR/+Y6cKubyFWVxSBG/NHztQc3+QC1zdkIxGNJ`<br>`bEHyFoajzNEO8LLOVFIMCZ2Td7gxUGRZc+fbqS33GZsCAX6v0=`<br>`bitcount:262144`<br>`fingerprint:`<br>`8d:44:ee:6c:ca:0b:44:95:36:d0:7d:f2:b5:78:74:7d`<br>`**********`<br>`could not retrieve dsa key information`<br>`**********` | Imports the keypair to the home directory of the switch.<br>The uri given here must be the uri of the private key and the public should be present on the same location with extension ".pub". The user will be prompted for the passphrase, and the same passphrase must be entered as was used to encrypt the key.<br>Once the private keys are copied to the switches which need to do passwordless copy to a server, and that server has the public key copied to its authorized_keys file in home directory, the user will be able to do passwordless file copy and ssh to the server from the switches.<br>**Note** To copy the public key to the authorized_keys file on the server, user can also copy the key from the show command mentioned above. |

| | Command | Purpose |
|---|---|---|
| Step 6 | `server# cat key_rsa.pub >> $HOME/.ssh/authorized_keys` | Appends the public key stored in key_rsa.pub to the authorized_keys file on the SCP server. The passwordless ssh/scp is then enabled from the switch to this server using the standard ssh and scp commands. |

# Recovering the Administrator Password

You can recover the administrator password using one of two methods:

- From the CLI with a user name that has network-admin privileges.
- Power cycling the switch.

The following topics included in this section:

- Using the CLI with Network-Admin Privileges, page 3-62
- Power Cycling the Switch, page 3-63

## Using the CLI with Network-Admin Privileges

If you are logged in to, or can log into, switch with a user name that has network-admin privileges and then recover the administrator password, follow these steps:

**Step 1** Use the **show user-accounts** command to verify that your user name has network-admin privileges.

```
switch# show user-account
user:admin
        this user account has no expiry date
        roles:network-admin

user:dbgusr
        this user account has no expiry date
        roles:network-admin network-operator
```

**Step 2** If your user name has network-admin privileges, issue the **username** command to assign a new administrator password.

```
switch# config t
switch(config)# username admin password <new password>
switch(config)# exit
switch#
```

**Step 3** Save the software configuration.

```
switch# copy running-config startup-config
```

# Power Cycling the Switch

If you cannot start a session on the switch that has network-admin privileges, you must recover the administrator password by power cycling the switch.

**Caution** This procedure disrupts all traffic on the switch. All connections to the switch will be lost for 2 to 3 minutes.

**Note** You cannot recover the administrator password from a Telnet or SSH session. You must have access to the local console connection. See the *Cisco MDS 9000 Family NX-OS Fundamentals Configuration Guide* for information on setting up the console connection.

To recover a administrator password by power cycling the switch, follow these steps:

**Step 1** For Cisco MDS 9500 Series switches with two supervisor modules, remove the supervisor module in slot 6 from the chassis.

**Note** On the Cisco MDS 9500 Series, the password recovery procedure must be performed on the active supervisor module. Removing the supervisor module in slot 6 ensures that a switchover will not occur during the password recovery procedure.

**Step 2** Power cycle the switch.

**Step 3** Press the **Ctrl-]** key sequence when the switch begins its Cisco NX-OS software boot sequence to enter the `switch(boot)#` prompt mode.

```
Ctrl-]
switch(boot)#
```

**Step 4** Change to configuration mode.

```
switch(boot)# config terminal
```

**Step 5** Issue the admin-password command to reset the administrator password. This will disable remote authentication for login through console, if enabled. This is done to ensure that admin is able to login through console with new password after password recovery. Telnet/SSH authentication will not be affected by this.

```
switch(boot-config)# admin-password <new password>
WARNING! Remote Authentication for login through console will be disabled#
```

For information on strong passwords, see the "Checking Password Strength" section on page 3-45.

**Step 6** Exit to the EXEC mode.

```
switch(boot-config)# admin-password <new password>
```

**Step 7** Issue the **load** command to load the Cisco NX-OS software.

```
switch(boot)# load bootflash:m9500-sf1ek9-mz.2.1.1a.bin
```

**Caution** If you boot a system image that is older than the image you used to store the configuration and do not use the **install all** command to boot the system, the switch erases the binary configuration and uses the ASCII configuration. When this occurs, you must use the **init system** command to recover your password.

**Step 8** Log in to the switch using the new administrator password.

```
switch login: admin
Password: <new password>
```

**Step 9** Reset the new password to ensure that is it is also the SNMP password for Fabric Manager.

```
switch# config t
switch(config)# username admin password <new password>
switch(config)# exit
switch#
```

**Step 10** Save the software configuration.

```
switch# copy running-config startup-config
```

**Step 11** Insert the previously removed supervisor module into slot 6 in the chassis.

# Default Settings

Table 3-3 lists the default settings for all switch security features in any switch.

*Table 3-3 Default Switch Security Settings*

| Parameters | Default |
|---|---|
| Roles in Cisco MDS Switches | Network operator (network-operator) |
| AAA configuration services | Local |
| Authentication port | 1821 |
| Accounting port | 1813 |
| Preshared key communication | Clear text |
| RADIUS server time out | 1 (one) second |
| RADIUS server retries | Once |
| TACACS+ | Disabled |
| TACACS+ servers | None configured |
| TACACS+ server timeout | 5 seconds |
| AAA server distribution | Disabled |
| VSAN policy for roles | Permit |
| User account | No expiry (unless configured) |
| Password | None |
| Password-strength | Enabled |

*Table 3-3 Default Switch Security Settings (continued)*

| Parameters | Default |
|---|---|
| Accounting log size | 250 KB |
| SSH service | Enabled |
| Telnet service | Disabled |


# Configuring Security Features on an External AAA Server

The authentication, authorization, and accounting (AAA) feature verifies the identity of, grants access to, and tracks the actions of users managing a switch. All Cisco MDS 9000 Family switches use Remote Access Dial-In User Service (RADIUS) or Terminal Access Controller Access Control device Plus (TACACS+) protocols to provide solutions using remote AAA servers.

Based on the user ID and password combination provided, switches perform local authentication or authorization using the local database or remote authentication or authorization using a AAA server. A preshared secret key provides security for communication between the switch and AAA servers. This secret key can be configured for all AAA servers or for only a specific AAA server. This security feature provides a central management capability for AAA servers.

This chapter includes the following sections:

# Switch Management Security

Management security in any switch in the Cisco MDS 9000 Family provides security to all management access methods, including the command-line interface (CLI) or Simple Network Management Protocol (SNMP).

This section includes the following topics:



## CLI Security Options

You can access the CLI using the console (serial connection), Telnet, or Secure Shell (SSH).

- Remote security control
  - Using RADIUS

    See the "Configuring RADIUS Server Monitoring Parameters" section on page 4-79
  - Using TACACS+

    See the "Configuring TACACS+ Server Monitoring Parameters" section on page 4-90
- Local security control.

  See the "Local AAA Services" section on page 4-110.

These security features can also be configured for the following scenarios:

- iSCSI authentication

  See the *Cisco MDS 9000 Family NX-OS IP Services Configuration Guide Cisco Fabric Manager IP Services Configuration Guide.*
- Fibre Channel Security Protocol (FC-SP) authentication

  See Chapter 8, "Configuring FC-SP and DHCHAP."

## SNMP Security Options

The SNMP agent supports security features for SNMPv1, SNMPv2c, and SNMPv3. Normal SNMP security features apply to all applications that use SNMP (for example, Cisco MDS 9000 Fabric Manager).

SNMP security options also apply to the Fabric Manager and Device Manager.

See the *Cisco MDS 9000 NX-OS Family System Management Configuration Guide* for more information on the SNMP security options.

Refer to the *Cisco Fabric Manager Fundamentals Configuration Guide* for information on Fabric Manager and Device Manager.

# Switch AAA Functionalities

Using the CLI or Fabric Manager, or an SNMP application, you can configure AAA switch functionalities on any switch in the Cisco MDS 9000 Family.

This section includes the following topics:



# Authentication

Authentication is the process of verifying the identity of the person or device accessing the switch. This identity verification is based on the user ID and password combination provided by the entity trying to access the switch. Cisco MDS 9000 Family switches allow you to perform local authentication (using the local lookup database) or remote authentication (using one or more RADIUS or TACACS+ servers).

**Note** Fabric Manager does not support AAA passwords with trailing white space, for example “passwordA.”

# Authorization

The following authorization roles exist in all Cisco MDS switches:

- Network operator (network-operator)—Has permission to view the configuration only. The operator cannot make any configuration changes.
- Network administrator (network-admin)— Has permission to execute all commands and make configuration changes. The administrator can also create and customize up to 64 additional roles.
- Default-role—Has permission to use the GUI (Fabric Manager and Device Manager). This access is automatically granted to all users for accessing the GUI.

These roles cannot be changed or deleted. You can create additional roles and configure the following options:

- Configure role-based authorization by assigning user roles locally or using remote AAA servers.
- Configure user profiles on a remote AAA server to contain role information. This role information is automatically downloaded and used when the user is authenticated through the remote AAA server.

**Note** If a user belongs only to one of the newly created roles and that role is subsequently deleted, then the user immediately defaults to the network-operator role.

# Accounting

The accounting feature tracks and maintains a log of every management configuration used to access the switch. This information can be used to generate reports for troubleshooting and auditing purposes. Accounting logs can be stored locally or sent to remote AAA servers.

# Remote AAA Services

Remote AAA services provided through RADIUS and TACACS+ protocols have the following advantages over local AAA services:

- User password lists for each switch in the fabric can be managed more easily.
- AAA servers are already deployed widely across enterprises and can be easily adopted.
- The accounting log for all switches in the fabric can be centrally managed.
- User role mapping for each switch in the fabric can be managed more easily.

# Remote Authentication Guidelines

If you prefer using remote AAA servers, follow these guidelines:

- A minimum of one AAA server should be IP reachable.
- Be sure to configure a desired local AAA policy as this policy is used if all AAA servers are not reachable.
- AAA servers are easily reachable if an overlay Ethernet LAN is attached to the switch (see the *Cisco Fabric Manager IP Services Configuration Guide* and the *Cisco MDS 9000 Family NX-OS Configuration Guide*). We recommend this method.
- SAN networks connected to the switch should have at least one gateway switch connected to the Ethernet LAN reaching the AAA servers.

# Server Groups

You can specify remote AAA servers for authentication, authorization, and accounting using server groups. A server group is a set of remote AAA servers implementing the same AAA protocol. The purpose of a server group is to provide for failover servers in case a remote AAA server fails to respond. If the first remote server in the group fails to respond, the next remote server in the group is tried until one of the servers sends a response. If all the AAA servers in the server group fail to respond, then that server group option is considered a failure. If required, you can specify multiple server groups. If the Cisco MDS switch encounters errors from the servers in the first group, it tries the servers in the next server group.

# AAA Service Configuration Options

AAA configuration in Cisco MDS 9000 Family switches is service based. You can have separate AAA configurations for the following services:

- Telnet or SSH login (Fabric Manager and Device Manager login)

- Console login
- iSCSI authentication (See the *Cisco Fabric Manager IP Services Configuration Guide* and the *Cisco MDS 9000 Family NX-OS IP Services Configuration Guide*)
- FC-SP authentication (See Chapter 8, "Configuring FC-SP and DHCHAP")
- Accounting

In general, server group, local, and none are the three options that can be specified for any service in an AAA configuration. Each option is tried in the order specified. If all the options fail, local is tried.

**Caution** Cisco MDS NX-OS supports user names that are created with alphanumeric characters or specific special characters (+ [plus], = [equal], _ [underscore], - [hyphen] , \ [backslash], and . [period]) whether created remotely (using TACACS+ or RADIUS) or locally, provided the user name starts with an alphabetical character. Local user names cannot be created with all numbers or with any special characters (apart from those specified). If a numeric-only user name or a non-supported special character user name exists on an AAA server, and is entered during login, then the user is denied access.

**Note** Even if local is not specified as one of the options, it is tried by default if all AAA servers configured for authentication are unreachable. User has the flexibility to disable this fallback.

When RADIUS times out, local login is attempted depending on the fallback configuration. For this local login to be successful, a local account for the user with the same password should exist, and the RADIUS timeout and retries should take less than 40 seconds. The user is authenticated if the username and password exist in the local authentication configuration.

Table 4-1 provides the related CLI command for each AAA service configuration option.

*Table 4-1 AAA Service Configuration Commands*

| AAA Service Configuration Option | Related Command |
|---|---|
| Telnet or SSH login (Cisco Fabric Manager and Device Manager login) | **aaa authentication login default** |
| Console login | **aaa authentication login console** |
| iSCSI authentication | **aaa authentication iscsi default** |
| FC-SP authentication | **aaa authentication dhchap default** |
| Accounting | **aaa accounting default** |

**Note** If we do not configure any authentication method for the console, the default authentication method will be applied for both console and Telnet or SSH.

# Error-Enabled Status

When you log in, the login is processed by rolling over to local user database if the remote AAA servers do not respond. In this situation, the following message is displayed on your screen if you have enabled the error-enabled feature:

```
Remote AAA servers unreachable; local authentication done.
```

To enable this message display, use the **aaa authentication login error-enable** command.

To disable this message display, use the **no aaa authentication login error-enable** command.

To view the current display status, use the **show aaa authentication login error-enable** command (see Example 4-1).

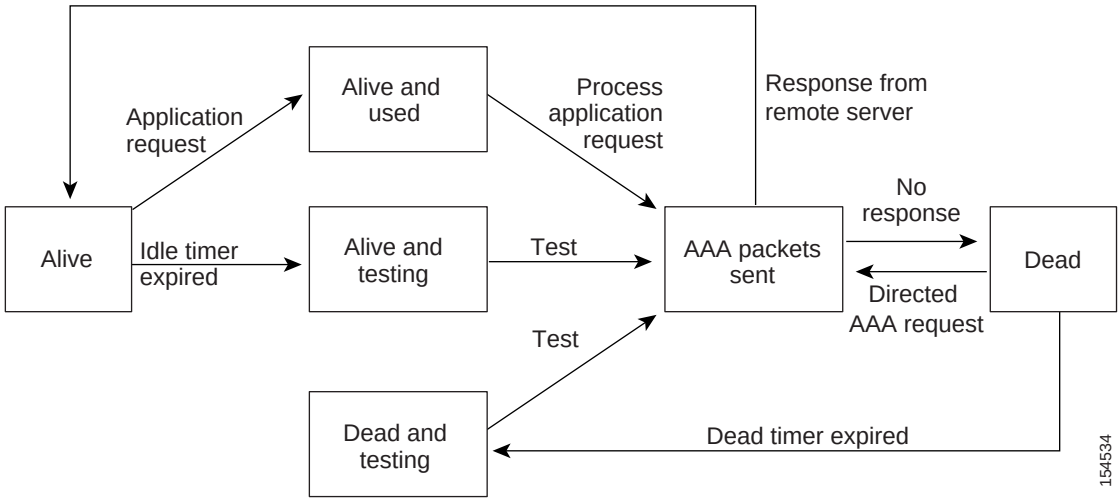
***Example 4-1 Displays AAA Authentication Login Information***

```
switch# show aaa authentication login error-enable
enabled
```

# AAA Server Monitoring

An unresponsive AAA server introduces a delay in the processing of AAA requests. An MDS switch can periodically monitor an AAA server to check whether it is responding (or alive) to save time in processing AAA requests. The MDS switch marks unresponsive AAA servers as dead and does not send AAA requests to any dead AAA servers. An MDS switch periodically monitors dead AAA servers and brings them to the alive state once they are responding. This monitoring process verifies that an AAA server is in a working state before real AAA requests are sent its way. Whenever an AAA server changes to the dead or alive state, an SNMP trap is generated and the MDS switch warns the administrator that a failure is taking place before it can impact performance. See Figure 4-1 for AAA server states.

***Figure 4-1 AAA Server States***




**Note** The monitoring interval for alive servers and dead servers is different and can be configured by the user. The AAA server monitoring is performed by sending a test authentication request to the AAA server.

The user name and password to be used in the test packet can be configured.

See the "Configuring RADIUS Server Monitoring Parameters" section on page 4-79"Configuring RADIUS Server Monitoring Parameters" section on page 4-83 and"Displaying RADIUS Server Details" section on page 4-88.

# Authentication and Authorization Process

Authentication is the process of verifying the identity of the person managing the switch. This identity verification is based on the user ID and password combination provided by the person managing the switch. The Cisco MDS 9000 Family switches allow you to perform local authentication (using the lookup database) or remote authentication (using one or more RADIUS servers or TACACS+ servers).

Authorization provides access control. It is the process of assembling a set of attributes that describe what the user is authorized to perform. Based on the user ID and password combination, the user is authenticated and authorized to access the network as per the assigned role. You can configure parameters that can prevent unauthorized access by an user, provided the switches use the TACACS+ protocol.

AAA authorization is the process of assembling a set of attributes that describe what the user is authorized to perform. Authorization in the Cisco NX-OS software is provided by attributes that are downloaded from AAA servers. Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights with the appropriate user.

The following steps explain the authorization and authentication process:

**Step 1** Log in to the required switch in the Cisco MDS 9000 Family, using the Telnet, SSH, Fabric Manager or Device Manager, or console login options.

**Step 2** When you have configured server groups using the server group authentication method, an authentication request is sent to the first AAA server in the group.

- If the AAA server fails to respond, then the next AAA server is contacted and so on until the remote server responds to the authentication request.
- If all AAA servers in the server group fail to respond, then the servers in the next server group are contacted.
- If all configured methods fail, then by default local database is used for authentication. The next section will describe the way to disable this fallback.

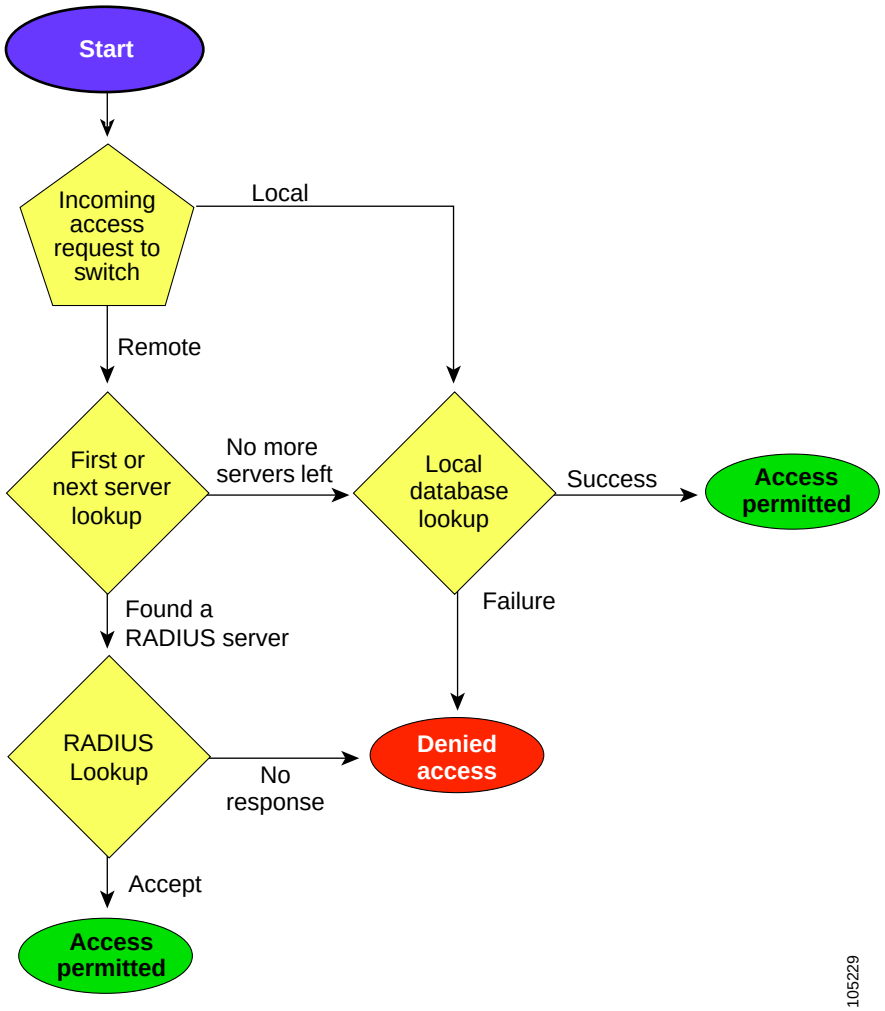
**Step 3** When you are successfully authenticated through a remote AAA server, then the following possible actions are taken:

- If the AAA server protocol is RADIUS, then user roles specified in the **cisco-av-pair** attribute are downloaded with an authentication response.
- If the AAA server protocol is TACACS+, then another request is sent to the same server to get the user roles specified as custom attributes for the shell.
- If user roles are not successfully retrieved from the remote AAA server, then the user is assigned the network-operator role if the **show aaa user default-role** command is enabled. You are denied access if this command is disabled.

**Step 4** When your user name and password are successfully authenticated locally, you are allowed to log in, and you are assigned the roles configured in the local database.

Figure 4-2 shows a flow chart of the authorization and authentication process.

*Figure 4-2 Switch Authorization and Authentication Flow*




**Note** No more server groups left = no response from any server in all server groups.
No more servers left = no response from any server within this server group.

To configure role-based authorization on TACACS+ server, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# aaa authorization` | Enables configuration of authorization methods. |
| **Step 3** | `switch(config)# aaa authorization config-commands` | Enables authorization for all commands under config mode Layer2 and Layer3. |
| **Step 4** | `switch(config)# aaa authorization config-commands default group tac1` | Enables specified TACACS+ server group authorization. |
| **Step 5** | `switch(config)# aaa authorization commands` | Enables AAA authorization for all EXEC mode commands. |

| | Command | Purpose |
|---|---|---|
| Step 6 | `switch(config)# aaa authorization commands default group tac1` | Enables specified TACACS+ server group authorization. |
| Step 7 | `switch(config)# aaa authorization commands default group local` | Enables default TACACS+ server group authorization.Authorization is based on the local-user-database. |
| Step 8 | `switch(config)# no aaa authorization command default group tac1` | Removes authorization for a specified function for the authenticated user. |

**Note**

- Authorization configuration is provided only for authentication done using TACACS+ server.
- The ‘none’ option from aaa authorization methods has been deprecated. If you did an upgrade from 4.x image and ‘none’ was configured as one of the authorization methods, it is be replaced with local. The functionality remains the same.
- Command authorization disables user role-based authorization control (RBAC), including the default roles.

You can use the **show** commands to display information on the AAA authorization and the default user roles assigned for remote authentication. (see Example 4-2 to Example 4-3).

*Example 4-2 Displays aaa Authorization Information Details*

```
switch# show aaa authorization all
AAA command authorization:
         default authorization for config-commands: local
         default authorization for commands: local
         cts: group rad1
```

*Example 4-3 Displays Default User Role for Remote Authentication*

```
switch# show aaa user default-role
enabled
```

## Configuring Fallback Mechanism for Authentication

You can enable/disable fallback to local database in case the remote authentication is set and all AAA servers are unreachable (authentication error). The fallback is set to local by default in case of an authentication error. You can disable this fallback for both console and ssh/telnet login. Disabling this fallback will tighten the security of authentication.

The CLI syntax and behavior is as follows:

| | Command | Purpose |
|---|---|---|
| Step 1 | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |

| | Command | Purpose |
|---|---|---|
| **Step 2** | `switch(config)# show run aaa all`<br>`aaa authentication login default fallback error local`<br>`aaa authentication login console fallback error local` | Displays the default fallback behavior. |
| **Step 3** | `switch(config)# no aaa authentication login default fallback error local`<br>`WARNING!!! Disabling fallback can lock your switch.` | Disables the fallback to local database for authentication.<br>**Note** Replace **default** with **console** in this command to disable fallback to console. |

**Caution** If fallback is disable for both default/console, remote authentication is enabled and servers are unreachable, then the switch will be locked.

## Verifying Authorization Profile

You can verify the authorizing profile for different commands. When enabled, all commands are directed to the Access Control Server (ACS) for verification. The verification details are displayed once the verification is completed.

```
switch# terminal verify-only username sikander
switch# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# feature telnet
% Success
switch(config)# feature ssh
% Success
switch(config)# end
% Success
switch# exit
```

**Note** This command only verifies the commands and does not enable the configuration.

## Testing Authorization

You can test the authorization settings for any command.

To test the authorization of a command, use the **test aaa authorization command-type** command.

```
switch(config)# test aaa authorization command-type commands user u1 command "feature dhcp"
% Success
```

# Configuring AAA Server Monitoring Parameters Globally

The AAA server monitoring parameters can be configured globally for all servers or individually for a specific server. This section explains how the global configuration can be set. The global configurations will apply to all servers that do not have individual monitoring parameters defined. For any server, the individual test parameter defined for that particular server will always get precedence over the global settings.

Use the following commands to configure the global monitoring parameters for RADIUS servers:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# radius-server deadtime 10` | Sets global deadtime for RADIUS servers to 10 minutes.<br>Acceptable Range: 0 to 1440 minutes. |
| **Step 3** | `switch(config)# radius-server timeout 20f` | Sets global timeout for RADIUS servers to 20 seconds.<br>Acceptable Range: 1 to 60 seconds. |
| **Step 4** | `switch(config)# radius-server retransmit 2` | Sets global retransmit count for RADIUS servers to 2.<br>Acceptable Range 0 to 5 |
| **Step 5** | `switch(config)# radius-server test username username password password idle-time time` | Globally configures test parameters for the RADIUS servers. |
| | `switch(config)# radius-server test username username password password no` | Disables global test parameters for the RADIUS servers. |

**Note** Replace “radius” with “tacacs” in the steps above to get equivalent commands for TACACS server global test parameter configurations.

The Global AAA Server Monitoring Parameters observe the following behavior:

- When a new AAA server is configured it is monitored using the global test parameters, if defined.
- When global test parameters are added or modified, all the AAA servers, which do not have any test parameters configured, start getting monitored using the new global test parameters.
- When the server test parameters are removed for a server or when the idle-time is set to zero (default value) it starts getting monitored using the global test parameters, if defined.
- If global test parameters are removed or global idle-time is set to zero, servers for which the server test parameters are present will not be affected. However monitoring will stop for all other servers which were previously being monitored using global parameters.
- If the server monitoring fails with the user specified server test parameters, the server monitoring does not fall back to global test parameters.

# Configuring LDAP

The Lightweight Directory Access Protocol (LDAP) provides centralized validation of users attempting to gain access to a Cisco NX-OS device. LDAP services are maintained in a database on an LDAP daemon running, typically, on a UNIX or Windows NT workstation. You must have access to and must configure an LDAP server before the configured LDAP features on your Cisco NX-OS device are available.

LDAP provides for separate authentication and authorization facilities. LDAP allows for a single access control server (the LDAP daemon) to provide each service-authentication and authorization-independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

The LDAP client/server protocol uses TCP (TCP port 389) for transport requirements. Cisco NX-OS devices provide centralized authentication using the LDAP protocol.

**Note** If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

This section includes the following topics:



# LDAP Authentication and Authorization

Clients establish a TCP connection and authentication session with an LDAP server through a simple bind (username and password). As part of the authorization process, the LDAP server searches its database to retrieve the user profile and other information.

You can configure the bind operation to first bind and then search, where authentication is performed first and authorization next, or to first search and then bind. The default method is to first search and then bind.

The advantage of searching first and binding later is that the distinguished name (DN) received in the search result can be used as the user DN during binding rather than forming a DN by prepending the username (cn attribute) with the baseDN. This method is especially helpful when the user DN is different from the username plus the baseDN. For the user bind, the bindDN is constructed as baseDN + append-with-baseDN, where append-with-baseDN has a default value of cn=$userid.

**Note** As an alternative to the bind method, you can establish LDAP authentication using the compare method, which compares the attribute values of a user entry at the server. For example, the user password attribute can be compared for authentication. The default password attribute type is userPassword.

# Guidelines and Limitations for LDAP

LDAP has the following guidelines and limitations:

- You can configure a maximum of 64 LDAP servers on the Cisco NX-OS device.
- Cisco NX-OS supports only LDAP version 3.
- Cisco NX-OS supports only these LDAP servers:
  - OpenLDAP
  - Microsoft Active Directory
- LDAP over Secure Sockets Layer (SSL) supports only SSL version 3 and Transport Layer Security (TLS) Version 1.0, Version 1.1, and Version 1.2.
- If you have a user account configured on the local Cisco NX-OS device that has the same name as a remote user account on an AAA server, the Cisco NX-OS software applies the user roles for the local user account to the remote user, not the user roles configured on the AAA server.
- A Cisco MDS switch will assign a local role to remote users when LDAP uses remote authentication protocol, if all the following conditions are met:
  - The remote username on the LDAP server has the same name as the local user on the Cisco MDS switch. (For example, “test” is the username on the AD server and “test” is the username created on the local Cisco MDS switch).
  - The LDAP server is configured as AAA authentication on the Cisco MDS switch.
  - The role assigned for the local user and the remote user is different.

  Consider the following example where the LDAP server has the username "test" which is a member of the AD group "testgroup". The Cisco MDS switch has a role configured with the name "testgroup" which has certain permit roles assigned to it. This role is created in the Cisco MDS switch for remote users who login into switch using LDAP. The Cisco MDS switch also has a local username "test" and it has "network-admin" as the assigned role. The Cisco MDS switch is configured for AAA authentication and uses LDAP as an authentication protocol. In this scenario, if a user logs into the Cisco MDS switch using the username "test", the switch authenticates the user using LDAP authentication (it uses the password of the user "test" created on the AD server). But, it assigns the role "network-admin", which is assigned to the local user "test", and not the "testgroup" role that is assigned to the remote authenticated user.

# Prerequisites for LDAP

LDAP has the following prerequisites:

- Obtain the IPv4 or IPv6 addresses or hostnames for the LDAP servers.
- Ensure that the Cisco NX-OS device is configured as an LDAP client of the AAA servers.

# Default Settings

Table 4-2 lists the default settings for LDAP parameters.

*Table 4-2 Default LDAP Parameter Settings*

| Parameters | Default |
|---|---|
| LDAP | Disabled |
| LDAP authentication method | First search and then bind |
| LDAP authentication mechanism | Plain |
| Dead-interval time | 0 minutes |
| Timeout interval | 5 seconds |
| Idle timer interval | 60 minutes |
| Periodic server monitoring username | test |
| Periodic server monitoring password | Cisco |

# Enabling LDAP

By default, the LDAP feature is disabled on the Cisco NX-OS device. You must explicitly enable the LDAP feature to access the configuration and verification commands for authentication.

To enable LDAP, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | `switch(config)# feature ldap` | Enables LDAP. |
| **Step 3** | `switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | `switch# copy running-config startup-config` | (Optional) Copies the running configuration to the startup configuration. |

# Configuring LDAP Server Hosts

To access a remote LDAP server, you must configure the IP address or the hostname for the LDAP server on the Cisco NX-OS device. You can configure up to 64 LDAP servers.

**Note** By default, when you configure an LDAP server IP address or hostname on the Cisco NX-OS device, the LDAP server is added to the default LDAP server group. You can also add the LDAP server to another LDAP server group.

To configure LDAP server hosts, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | `switch(config)# ldap-server host 10.10.2.2 enable-ssl` | Specifies the IPv4 or IPv6 address or hostname for an LDAP server.<br>The **enable-ssl** keyword ensures the integrity and confidentiality of the transferred data by causing the LDAP client to establish a Secure Sockets Layer (SSL) session prior to sending the bind or search request. |
| **Step 3** | `switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | `switch# copy running-config startup-config` | (Optional) Copies the running configuration to the startup configuration. |

# Configuring the RootDN for an LDAP Server

You can configure the root designated name (DN) for the LDAP server database. The rootDN is used to bind to the LDAP server to verify its state.

To configure the RootDN for an LDAP server, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | `switch(config)# ldap-server host 10.10.1.1 rootDN cn=manager,dc=acme,dc=com password Ur2Gd2BH timeout 60` | Specifies the rootDN for the LDAP server database and the bind password for the root.<br>Optionally specifies the TCP port to use for LDAP messages to the server. The range is from 1 to 65535, and the default TCP port is the global value or 389 if a global value is not configured. Also specifies the timeout interval for the server. The range is from 1 to 60 seconds, and the default timeout is the global value or 5 seconds if a global value is not configured. |
| **Step 3** | `switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | `switch# show ldap-server` | (Optional) Displays the LDAP server configuration. |
| **Step 5** | `switch# copy running-config startup-config` | (Optional) Copies the running configuration to the startup configuration. |

# Configuring LDAP Server Groups

You can specify one or more remote AAA servers to authenticate users using server groups. All members of a group must be configured to use LDAP. The servers are tried in the same order in which you configure them.

You can configure these server groups at any time, but they take effect only when you apply them to an AAA service.

Starting from Cisco MDS NX-OS Release 6.2(1), Cisco MDS 9000 Series switches support group-based user roles. You can create a group on the LDAP servers and also create a group with the exact same name on the Cisco MDS switch and then add users to the group. The user role attribute is inherited by the user from the group that is configured. This can be accomplished using the Microsoft LDAP Server's built-in memberOf attribute. If you wish to use the memberOf attribute, ensure that you create a role name on the switch. The role name must be the same as the group name on the LDAP server.

**Note**

- A user can be part of only one group that is available on the switch.
- A user can be part of multiple groups, but only one group should be part of the switch role.
- A group name cannot have a space.

To configure the LDAP server groups, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | `switch(config)# aaa group server ldap LDAPServer1`<br>`switch(config-ldap)#` | Creates an LDAP server group and enters the LDAP server group configuration mode for that group. |
| **Step 3** | `switch(config-ldap)# server 10.10.2.2` | Configures the LDAP server as a member of the LDAP server group.<br>If the specified LDAP server is not found, configure it using the ldap-server host command and retry this command. |
| **Step 4** | `switch(config-ldap)# authentication compare password-attribute TyuL8r` | (Optional) Performs LDAP authentication using the bind or compare method. The default LDAP authentication method is the bind method using first search and then bind. |
| **Step 5** | `switch(config-ldap)# enable user-server-group` | (Optional) Enables group validation. The group name should be configured in the LDAP server. Users can login through public-key authentication only if the username is listed as a member of this configured group in the LDAP server. |
| **Step 6** | `switch(config-ldap)# enable Cert-DN-match` | (Optional) Enables users to login only if the user profile lists the subject-DN of the user certificate as authorized for login. |
| **Step 7** | `switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 8** | `switch# show ldap-server groups` | (Optional) Displays the LDAP server group configuration. |
| **Step 9** | `switch# show run ldap` | (Optional) Displays the LDAP configuration. |
| **Step 10** | `switch# copy running-config startup-config` | (Optional) Copies the running configuration to the startup configuration. |

# Configuring the Global LDAP Timeout Interval

You can set a global timeout interval that determines how long the Cisco NX-OS device waits for responses from all LDAP servers before declaring a timeout failure.

To configure the global LDAP timeout interval, follow these steps:

| | | |
|---|---|---|
| **Step 1** | `switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | `switch(config)# ldap-server timeout 10` | Specifies the timeout interval for LDAP servers. The default timeout interval is 5 seconds. The range is from 1 to 60 seconds. |
| **Step 3** | `switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | `switch# show ldap-server` | (Optional) Displays the LDAP server configuration. |
| **Step 5** | `switch# copy running-config startup-config` | (Optional) Copies the running configuration to the startup configuration. |

# Configuring the Timeout Interval for an LDAP Server

You can set a timeout interval that determines how long the Cisco NX-OS device waits for responses from an LDAP server before declaring a timeout failure.

To configure the timeout interval for an LDAP server, follow these steps:

| | | |
|---|---|---|
| **Step 1** | `switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | `switch(config)# ldap-server host server1 timeout 10` | Specifies the timeout interval for a specific server. The default is the global value.<br>**Note** The timeout interval value specified for an LDAP server overrides the global timeout interval value specified for all LDAP servers. |
| **Step 3** | `switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | `switch# show ldap-server` | (Optional) Displays the LDAP server configuration. |
| **Step 5** | `switch# copy running-config startup-config` | (Optional) Copies the running configuration to the startup configuration. |

# Configuring the Global LDAP Server Port

You can configure a global LDAP server port through which clients initiate TCP connections. By default, Cisco NX-OS devices use port 389 for all LDAP requests.

To configure the global LDAP server port, follow these steps:

| | | |
|---|---|---|
| **Step 1** | `switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | `switch(config)# ldap-server port 2` | Specifies the global TCP port to use for LDAP messages to the server. The default TCP port is 389. The range is from 1 to 65535. |
| **Step 3** | `switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | `switch# show ldap-server` | (Optional) Displays the LDAP server configuration. |
| **Step 5** | `switch# copy running-config startup-config` | (Optional) Copies the running configuration to the startup configuration. |

# Configuring TCP Ports

You can configure another TCP port for the LDAP servers if there are conflicts with another application. By default, Cisco NX-OS devices use port 389 for all LDAP requests.

To configure the TCP ports, follow these steps:

| | | |
|---|---|---|
| **Step 1** | `switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | `switch(config)# ldap-server host 10.10.1.1 port 200 timeout 5` | Specifies the TCP port to use for LDAP messages to the server. The default TCP port is 389. The range is from 1 to 65535. Optionally specifies the timeout interval for the server. The range is from 1 to 60 seconds, and the default timeout is the global value or 5 seconds if a global value is not configured.<br>**Note** The timeout interval value specified for an LDAP server overrides the global timeout interval value specified for all LDAP servers. |
| **Step 3** | `switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | `switch# show ldap-server` | (Optional) Displays the LDAP server configuration. |
| **Step 5** | `switch# copy running-config startup-config` | (Optional) Copies the running configuration to the startup configuration. |

# Configuring LDAP Search Maps

You can configure LDAP search maps to send a search query to the LDAP server. The server searches its database for data meeting the criteria specified in the search map.

To configure the LDAP search maps, follow these steps:

| | | |
|---|---|---|
| **Step 1** | `switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | `switch(config)# ldap search-map map1`<br>`switch(config-ldap-search-map)#` | Configures an LDAP search map. |
| **Step 3** | `Example 1:`<br>`switch(config-ldap-search-map)# userprofile attribute-name description search-filter "(&(objectClass=inetOrgPerson)(cn=$userid))" base-DN dc=acme,dc=com`<br>`Example 2:`<br>`switch(config-ldap-search-map)# userprofile attribute-name "memberOf" search-filter "(&(objectClass=inetOrgPerson)(cn=$userid))" base-DN dc=acme,dc=com` | (Optional) Configures the attribute name, search filter, and base-DN for the user profile, trusted certificate, CRL, certificate DN match, public key match, or user-switchgroup lookup search operation. These values are used to send a search query to the LDAP server.<br>**Note** The LDAP search filter string is limited to a maximum of 128 characters.<br>Specifies the groups to which the user is a member of. |
| **Step 4** | `switch(config-ldap-search-map)# exit`<br>`switch(config)#` | Exits LDAP search map configuration mode. |
| **Step 5** | `switch(config)# show ldap-search-map` | (Optional) Displays the configured LDAP search maps. |
| **Step 6** | `switch# copy running-config startup-config` | (Optional) Copies the running configuration to the startup configuration. |

# Configuring the LDAP Dead-Time Interval

You can configure the dead-time interval for all LDAP servers. The dead-time interval specifies the time that the Cisco NX-OS device waits, after declaring that an LDAP server is dead, before sending out a test packet to determine if the server is now alive.

**Note** When the dead-time interval is 0 minutes, LDAP servers are not marked as dead even if they are not responding. You can configure the dead-time interval per group.

To configure the LDAP dead-time interval, follow these steps:

| | | |
|---|---|---|
| **Step 1** | `switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | `switch(config)# ldap-server deadtime 5` | Configures the global dead-time interval. The default value is 0 minutes. The range is from 1 to 60 minutes. |
| **Step 3** | `switch(config)# exit`<br>`switch#` | Exits configuration mode. |

| | | |
|---|---|---|
| **Step 4** | `switch# show ldap-server` | (Optional) Displays the LDAP server configuration. |
| **Step 5** | `switch# copy running-config startup-config` | (Optional) Copies the running configuration to the startup configuration. |

# Configuring AAA Authorization on LDAP Servers

You can configure the default AAA authorization method for LDAP servers.

To configure the AAA authorization on LDAP servers, follow these steps:

| | | |
|---|---|---|
| **Step 1** | `switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | `switch(config)# aaa authorization ssh-certificate default group LDAPServer1 LDAPServer2` | Configures the default AAA authorization method for the LDAP servers.<br>The **ssh-certificate** keyword configures LDAP or local authorization with certificate authentication, and the ssh-publickey keyword configures LDAP or local authorization with the **SSH public key**. The default authorization is local authorization, which is the list of authorized commands for the user's assigned role.<br>The group-list argument consists of a space-delimited list of LDAP server group names. Servers that belong to this group are contacted for AAA authorization. The `local` method uses the local database for authorization. |
| **Step 3** | `switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | `switch(config)# show aaa authorization` | (Optional) Displays the AAA authorization configuration. The all keyword displays the default values. |
| **Step 5** | `switch(config)# copy running-config startup-config` | (Optional) Copies the running configuration to the startup configuration. |

# Disabling LDAP

When you disable LDAP, all related configurations are automatically discarded.

To disable LDAP, follow these steps:

| | | |
|---|---|---|
| **Step 1** | `switch# configure terminal`<br>`switch(config)#` | Enters global configuration mode. |
| **Step 2** | `switch(config)# no feature ldap` | Disables LDAP. |
| **Step 3** | `switch(config)# exit`<br>`switch#` | Exits configuration mode. |
| **Step 4** | `switch# copy running-config startup-config` | (Optional) Copies the running configuration to the startup configuration. |

For detailed information about the fields in the output from this command, see the *Cisco MDS 9000 Family Command Reference, Release 5.0(1a)*.

# Configuration Examples for LDAP

The following example shows how to configure an LDAP server host and server group:

```
feature ldap
ldap-server host 10.10.2.2 enable-ssl
aaa group server ldap LdapServer
    server 10.10.2.2
exit
show ldap-server
show ldap-server groups
```

The following example shows how to configure an LDAP search map:

```
ldap search-map s0
userprofile attribute-name description search-filter
(&(objectClass=inetOrgPerson)(cn=$userid)) base-DN dc=acme,dc=com
exit
show ldap-search-map
```

The following example shows how to configure AAA authorization with certificate authentication for an LDAP server:

```
aaa authorization ssh-certificate default group LDAPServer1 LDAPServer2
exit
show aaa authorization
```

# Configuring RADIUS Server Monitoring Parameters

Cisco MDS 9000 Family switches can use the RADIUS protocol to communicate with remote AAA servers. You can configure multiple RADIUS servers and server groups and set timeout and retry counts.

RADIUS is a distributed client/server protocol that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco MDS 9000 Family switches and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

This section defines the RADIUS operation, identifies its network environments, and describes its configuration possibilities.

- Allowing Users to Specify a RADIUS Server at Login, page 4-86

# About RADIUS Server Default Configuration

Fabric Manager allows you to set up a default configuration that can be used for any RADIUS server that you configure the switch to communicate with. The default configuration includes:

- Encryption type
- Timeout value
- Number of retransmission attempts

- Allowing the user to specify a RADIUS server at login

# Setting the RADIUS Server Address

You can add up to 64 RADIUS servers. RADIUS keys are always stored in encrypted form in persistent storage. The running configuration also displays encrypted keys.

To specify the host RADIUS server IPv4 address and other options, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# radius-server host 10.10.0.0 key HostKey` | Specifies the preshared key for the selected RADIUS server. This key overrides the key assigned using the **radius-server key** command. In this example, the host is 10.10.0.0 and the key is HostKey. |
| **Step 3** | `switch(config)# radius-server host 10.10.0.0 auth-port 2003` | Specifies the destination UDP port number to which the RADIUS authentication messages should be sent. In this example, the host is 10.10.0.0 and the authentication port is 2003. The default authentication port is 1812, and the valid range is 0 to 65366. |
| **Step 4** | `switch(config)# radius-server host 10.10.0.0 acct-port 2004` | Specifies the destination UDP port number to which RADIUS accounting messages should be sent. The default accounting port is 1813, and the valid range is 0 to 65366. |
| **Step 5** | `switch(config)# radius-server host 10.10.0.0 accounting` | Specifies this server to be used only for accounting purposes.<br>**Note** If neither the **authentication** nor the **accounting** options are specified, the server is used for both accounting and authentication purposes. |
| **Step 6** | `switch(config)# radius-server host 10.10.0.0 key 0 abcd` | Specifies a clear text key for the specified server. The key is restricted to 64 characters. |
| | `switch(config)# radius-server host 10.10.0.0 key 4 da3Asda2ioyuoiuH` | Specifies an encrypted key for the specified server. The key is restricted to 64 characters. |

To specify the host RADIUS server IPv6 address and other options, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# radius-server host 2001:0DB8:800:200C::417A Key HostKey` | Specifies the preshared key for the selected RADIUS server. This key overrides the key assigned using the **radius-server key** command. In this example, the host is 2001:0DB8:800:200C::417A and the key is HostKey. |

| | Command | Purpose |
|---|---|---|
| **Step 3** | `switch(config)# radius-server host 2001:0DB8:800:200C::417A auth-port 2003` | Specifies the destination UDP port number to which the RADIUS authentication messages should be sent. In this example, the host is 2001:0DB8:800:200C::417A and the authentication port is 2003. The default authentication port is 1812, and the valid range is 0 to 65366. |
| **Step 4** | `switch(config)# radius-server host 2001:0DB8:800:200C::417A acct-port 2004` | Specifies the destination UDP port number to which RADIUS accounting messages should be sent. The default accounting port is 1813, and the valid range is 0 to 65366. |
| **Step 5** | `switch(config)# radius-server host 2001:0DB8:800:200C::417A accounting` | Specifies this server to be used only for accounting purposes.<br>**Note** If neither the **authentication** nor the **accounting** options are specified, the server is used for both accounting and authentication purposes. |
| **Step 6** | `switch(config)# radius-server host 2001:0DB8:800:200C::417A key 0 abcd` | Specifies a clear text key for the specified server. The key is restricted to 64 characters. |
| | `switch(config)# radius-server host 2001:0DB8:800:200C::417A key 4 da3Asda2ioyuoiuH` | Specifies an encrypted key for the specified server. The key is restricted to 64 characters. |

To specify the host RADIUS server DNS name and other options, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# radius-server host radius2 key HostKey` | Specifies the preshared key for the selected RADIUS server. This key overrides the key assigned using the **radius-server key** command. In this example, the host is radius2 and the key is HostKey. |
| **Step 3** | `switch(config)# radius-server host radius2 auth-port 2003` | Specifies the destination UDP port number to which the RADIUS authentication messages should be sent. In this example, the host is radius2 and the authentication port is 2003. The default authentication port is 1812, and the valid range is 0 to 65366. |
| **Step 4** | `switch(config)# radius-server host radius2 acct-port 2004` | Specifies the destination UDP port number to which RADIUS accounting messages should be sent. The default accounting port is 1813, and the valid range is 0 to 65366. |

| | Command | Purpose |
|---|---|---|
| **Step 5** | `switch(config)# radius-server host radius2 accounting` | Specifies this server to be used only for accounting purposes.<br>**Note** If neither the **authentication** nor the **accounting** options are specified, the server is used for both accounting and authentication purposes. |
| **Step 6** | `switch(config)# radius-server host radius2 key 0 abcd` | Specifies a clear text key for the specified server. The key is restricted to 64 characters. |
| | `switch(config)# radius-server host radius2 key 4 da3Asda2ioyuoiuH` | Specifies an encrypted key for the specified server. The key is restricted to 64 characters. |

# About the Default RADIUS Server Encryption Type and Preshared Key

You need to configure the RADIUS preshared key to authenticate the switch to the RADIUS server. The length of the key is restricted to 64 characters and can include any printable ASCII characters (white spaces are not allowed). You can configure a global key to be used for all RADIUS server configurations on the switch.

You can override this global key assignment by explicitly using the **key** option when configuring an individual RADIUS server in the **radius-server host** command.

# Configuring the Default RADIUS Server Encryption Type and Preshared Key

To configure the RADIUS preshared key, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# radius-server key AnyWord` | Configures a preshared key (AnyWord) to authenticate communication between the RADIUS client and server. The default is clear text. |
| | `switch(config)# radius-server key 0 AnyWord` | Configures a preshared key (AnyWord) specified in clear text (indicated by 0) to authenticate communication between the RADIUS client and server. |
| | `switch(config)# radius-server key 7 abe4DFeeweo00o` | Configures a preshared key (specified in encrypted text) specified in encrypted text (indicated by 7) to authenticate communication between the RADIUS client and server. |

# Setting the RADIUS Server Timeout Interval

You can configure a global timeout value between transmissions for all RADIUS servers.

**Note** If timeout values are configured for individual servers, those values override the globally configured values.

To specify the timeout values between retransmissions to the RADIUS servers, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# radius-server timeout 30` | Configures the global timeout period in seconds for the switch to wait for a response from all RADIUS+ servers before the switch declares a timeout failure. The time ranges from 1 to 1440 seconds. |
| | `switch(config)# no radius-server timeout 30` | Reverts the transmission time to the default value (1 second). |

# Setting the Default RADIUS Server Timeout Interval and Retransmits

By default, a switch retries transmission to a RADIUS server only once before reverting to local authentication. You can increase this number up to a maximum of five retries per server. You can also configure the timeout value for the RADIUS server.

To specify the number of times that RADIUS servers should try to authenticate a user, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# radius-server retransmit 3` | Configures the number of times (3) the switch tries to connect to a RADIUS server(s) before reverting to local authentication. |
| | `switch(config)# no radius-server retransmit` | Reverts to the default retry count (1). |

# Configuring RADIUS Server Monitoring Parameters

You can configure parameters for monitoring RADIUS servers. You can configure this option to test the server periodically, or you can run a one-time only test.

This section includes the following topics:



## Configuring the Test Idle Timer

The test idle timer specifies the interval during which a RADIUS server receives no requests before the MDS switch sends out a test packet.

**Note** The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, periodic RADIUS server monitoring is not performed.

To configure the idle timer, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# radius-server host 10.1.1.1 test idle-time 20` | Configures the test idle time interval value in minutes. The valid range is 1 to 1440 minutes. |
| **Step 3** | `switch(config)# no radius-server host 10.1.1.1 test idle-time 20` | Reverts to the default value (0 minutes). |

## Configuring Test User Name

You can configure a username and password for periodic RADIUS server status testing. You do not need to configure the test username and password to issue test messages to monitor RADIUS servers. You can use the default test username (test) and default password (test).

**Note** We recommend that the test username not be the same as an existing username in the RADIUS database for security reasons.

To configure the optional username and password for periodic RADIUS server status testing, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# radius-server host 10.1.1.1 test username testuser` | Configures the test user (testuser) with the default password (test). The default user name is test. |
| | `switch(config)# no radius-server host 10.1.1.1 test username testuser` | Removes the test user name (testuser). |
| | `switch(config)# radius-server host 10.1.1.1 test username testuser password Ur2Gd2BH` | Configures the test user (testuser) and assigns a strong password. |

## Configuring the Dead Timer

The dead timer specifies the interval that the MDS switch waits, after declaring that a RADIUS server is dead, before sending out a test packet to determine if the server is now alive.

**Note** The default dead timer value is 0 minutes. When the dead timer interval is 0 minutes, RADIUS server monitoring is not performed unless the RADIUS server is part of a server group and the dead-time interval for the group is greater than 0 minutes. (See the "Server Groups" section on page 4-62).

**Note** If the dead timer of a dead RADIUS server expires before it is sent a RADIUS test message, that server is marked as alive again even if it is still not responding. To avoid this scenario, configure a test user with a shorter idle time than the dead timer time.

To configure the dead timer, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# radius-server deadtime 30` | Configures the dead timer interval value in minutes. The valid range is 1 to 1440 minutes. |
| **Step 3** | `switch(config)# no radius-server deadtime 30` | Reverts to the default value (0 minutes). |

# About RADIUS Servers

You can add up to 64 RADIUS servers. RADIUS keys are always stored in encrypted form in persistent storage. The running configuration also displays encrypted keys. When you configure a new RADIUS server, you can use the default configuration or modify any of the parameters to override the default RADIUS configuration.

## Configuring the Test Idle Timer

The test idle timer specifies the interval during which a RADIUS server receives no requests before the MDS switch sends out a test packet.

**Note** The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, periodic RADIUS server monitoring is not performed.

To configure the test idle timer, see Configuring RADIUS Server Monitoring Parameters, page 4-79.

## Configuring Test User Name

You can configure a username and password for periodic RADIUS server status testing. You do not need to configure the test username and password to issue test messages to monitor RADIUS servers. You can use the default test username (test) and default password (test).

**Note** We recommend that the test username not be the same as an existing username in the RADIUS database for security reasons.

To configure the optional username and password for periodic RADIUS server status testing, see Configuring RADIUS Server Monitoring Parameters, page 4-79.

# About Validating a RADIUS Server

As of Cisco SAN-OS Release 3.0(1), you can periodically validate a RADIUS server. The switch sends a test authentication to the server using the username and password that you configure. If the server does not respond to the test authentication, then the server is considered non responding.

**Note** For security reasons we recommend that you do not use a username that is configured on your RADIUS server as a test username.

You can configure this option to test the server periodically, or you can run a one-time only test.

# Sending RADIUS Test Messages for Monitoring

You can manually send test messages to monitor a RADIUS server.

To send the test message to the RADIUS server, follow this step:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# test aaa server radius 10.10.1.1 test test` | Sends a test message to a RADIUS server using the default username (test) and password (test). |
| | `switch# test aaa server radius 10.10.1.1 testuser Ur2Gd2BH` | Sends a test message to a RADIUS server using a configured test username (testuser) and password (Ur2Gd2BH).<br>**Note** A configured username and password is optional (see the "Configuring Test User Name" section on page 4-84). |

# Allowing Users to Specify a RADIUS Server at Login

By default, an MDS switch forwards an authentication request to the first server in the RADIUS server group. You can configure the switch to allow the user to specify which RADIUS server to send the authenticate request by enabling the directed request option. If you enable this option, the user can log in as *username@hostname*, where the *hostname* is the name of a configured RADIUS server.

**Note** User specified logins are supported only for Telnet sessions.

To allow users logging into an MDS switch to select a RADIUS server for authentication, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# radius-server directed-request` | Allows users to specify a RADIUS server to send the authentication request when logging in. |
| | `switch(config)# no radius-server directed-request` | Reverts to sending the authentication request to the first server in the server group (default). |

You can use the **show tacacs-server directed-request** command to display the RADIUS directed request configuration.

```
switch# show radius-server directed-request
disabled
```

# About Vendor-Specific Attributes

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific attributes (VSAs) between the network access server and the RADIUS server. The IETF uses attribute 26. VSAs allow vendors to support their own extended attributes that are not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. The Cisco vendor ID is 9, and the supported option is vendor type 1, which is named **cisco-avpair.** The value is a string with the following format:

```
protocol : attribute separator value *
```

Where **protocol** is a Cisco attribute for a particular type of authorization, **separator** is `=` (equal sign) for mandatory attributes, and `*` (asterisk) is for optional attributes.

When you use RADIUS servers to authenticate yourself to a Cisco MDS 9000 Family switch, the RADIUS protocol directs the RADIUS server to return user attributes, such as authorization information, along with authentication results. This authorization information is specified through VSAs.

## VSA Format

The following VSA protocol options are supported by the Cisco NX-OS software:

- **Shell** protocol—Used in Access-Accept packets to provide user profile information.
- **Accounting** protocol—Used in Accounting-Request packets. If a value contains any white spaces, it should be put within double quotation marks.

The following attributes are supported by the Cisco NX-OS software:

- **roles**—This attribute lists all the roles to which the user belongs. The value field is a string storing the list of group names delimited by white space. For example, if you belong to roles **vsan-admin** and **storage-admin**, the value field would be "**vsan-admin storage-admin"**. This subattribute is sent in the VSA portion of the Access-Accept frames from the RADIUS server, and it can only be used with the shell protocol value. These are two examples using the roles attribute:

  ```
  shell:roles="network-admin vsan-admin"
  shell:roles*"network-admin vsan-admin"
  ```

  When an VSA is specified as **shell:roles*"network-admin vsan-admin"**, this VSA is flagged as an optional attribute, and other Cisco devices ignore this attribute.

- **accountinginfo**—This attribute stores additional accounting information besides the attributes covered by a standard RADIUS accounting protocol. This attribute is only sent in the VSA portion of the Account-Request frames from the RADIUS client on the switch, and it can only be used with the accounting protocol-related PDUs.

## Specifying SNMPv3 on AAA Servers

The vendor/custom attribute **cisco-av-pair** can be used to specify user's role mapping using the format:

```
shell:roles="roleA roleB …"
```

**Note** When you log in to a Cisco MDS switch successfully using the Fabric Manager or Device Manager through Telnet or SSH and if that switch is configured for AAA server-based authentication, a temporary SNMP user entry is automatically created with an expiry time of one day. The switch authenticates the SNMPv3 protocol data units (PDUs) with your Telnet or SSH login name as the SNMPv3 user. The management station can temporarily use the Telnet or SSH login name as the SNMPv3 **auth** and **priv** passphrase. This temporary SNMP login is only allowed if you have one or more active MDS shell sessions. If you do not have an active session at any given time, your login is deleted and you will not be allowed to perform SNMPv3 operations.

If the role option in the **cisco-av-pair** attribute is not set, the default user role is network-operator.

The VSA format optionally specifies your SNMPv3 authentication and privacy protocol attributes also as follows:

```
shell:roles="roleA roleB..." snmpv3:auth=SHA priv=AES-128
```

The SNMPv3 authentication protocol options are SHA and MD5. The privacy protocol options are AES-128 and DES. If these options are not specified in the **cisco-av-pair** attribute on the ACS server, MD5 and DES are used by default.

# Displaying RADIUS Server Details

Use the **show radius-server** command to display configured RADIUS parameters as shown in Example 4-4.

***Example 4-4 Displays Configured RADIUS Information***

```
switch# show radius-server
Global RADIUS shared secret:*******
retransmission count:5
timeout value:10
following RADIUS servers are configured:
        myradius.cisco.users.com:
                available for authentication on port:1812
                available for accounting on port:1813
        172.22.91.37:
                available for authentication on port:1812
                available for accounting on port:1813
                RADIUS shared secret:******
        10.10.0.0:
                available for authentication on port:1812
                available for accounting on port:1813
                RADIUS shared secret:******
```

***Example 4-5 Displays Configured RADIUS Server-Group Order***

```
switch# show radius-server groups
total number of groups:4
following RADIUS server groups are configured:
```

```
        group radius:
                server: all configured radius servers
        group Group1:
                server: Server3 on auth-port 1812, acct-port 1813
                server: Server5 on auth-port 1812, acct-port 1813
        group Group5:
```

# Displaying RADIUS Server Statistics

You can display RADIUS server statistics using the **show radius-server statistics** command.

***Example 4-6 Displays RADIUS Server Statistics***

```
switch# show radius-server statistics 10.1.3.2
Server is not monitored

Authentication Statistics
        failed transactions: 0
        sucessful transactions: 0
        requests sent: 0
        requests timed out: 0
        responses with no matching requests: 0
        responses not processed: 0
        responses containing errors: 0

Accounting Statistics
        failed transactions: 0
        successful transactions: 0
        requests sent: 0
        requests timed out: 0
        responses with no matching requests: 0
        responses not processed: 0
        responses containing errors:
```

You can clear RADIUS server statistics using the **clear radius-server statistics 10.1.3.2** command.

# One-Time Password Support

A one-time password (OTP) is a password that is valid for a single login session or transaction. OTPs avoid a number of disadvantages that are associated with usual (static) passwords. The most vital disadvantage that is addressed by OTPs is that, they are not at risk to replay attacks. If an intruder manages to record an OTP that was already used to log into a service or to conduct an operation, it will not be misused as it will no longer be valid.

One Time Password is applicable only to RADIUS and TACACS protocol daemons. With a RADIUS protocol daemon, there is no configuration required from the switch side. With a TACACS protocol, ascii authentication mode needs to be enabled, which can be done by the following command:

```
aaa authentication login ascii-authentication
```

# Configuring TACACS+ Server Monitoring Parameters

A Cisco MDS switch uses the Terminal Access Controller Access Control System Plus (TACACS+) protocol to communicate with remote AAA servers. You can configure multiple TACACS+ servers and set timeout values.

This section includes the following topics:



## About TACACS+

TACACS+ is a client/server protocol that uses TCP (TCP port 49) for transport requirements. All switches in the Cisco MDS 9000 Family provide centralized authentication using the TACACS+ protocol. The TACACS+ has the following advantages over RADIUS authentication:

- Provides independent, modular AAA facilities. Authorization can be done without authentication.
- Uses the TCP transport protocol to send data between the AAA client and server, making reliable transfers with a connection-oriented protocol.
- Encrypts the entire protocol payload between the switch and the AAA server to ensure higher data confidentiality. The RADIUS protocol only encrypts passwords.

# About TACACS+ Server Default Configuration

Fabric Manager allows you to set up a default configuration that can be used for any TACACS+ server that you configure the switch to communicate with. The default configuration includes:

- Encryption type
- Preshared key
- Timeout value
- Number of retransmission attempts
- Allowing the user to specify a TACACS+ server at login

# About the Default TACACS+ Server Encryption Type and Preshared Key

You need to configure the TACACS+ preshared key to authenticate the switch to the TACACS+ server. The length of the key is restricted to 64 characters and can include any printable ASCII characters (white spaces are not allowed). You can configure a global key to be used for all TACACS+ server configurations on the switch.

You can override this global key assignment by explicitly using the **key** option when configuring and individual TACACS+ server.

# Enabling TACACS+

By default, the TACACS+ feature is disabled in all switches in the Cisco MDS 9000 Family. You must explicitly enable the TACACS+ feature to access the configuration and verification commands for fabric authentication. When you disable this feature, all related configurations are automatically discarded.

To enable TACACS+ for a Cisco MDS switch, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# feature tacacs+` | Enables the TACACS+ in this switch. |
| | `switch(config)# no feature tacacs+` | Disables (default) the TACACS+ in this switch. |

# Setting the TACACS+ Server Address

If a secret key is not configured for a configured server, a warning message is issued if a global key is not configured. If a server key is not configured, the global key (if configured) is used for that server (see the "Setting the Default TACACS+ Server Timeout Interval and Retransmits" section on page 4-93).

**Note** You can use the dollar sign ($) and the percent sign (%) in global secret keys.

To configure the TACACS+ server IPv4 address and other options, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# tacacs-server host 171.71.58.91` | Configures the TACACS+ server identified by the specified IPv4 address. |
| | `switch(config)# no tacacs-server host 171.71.58.91` | Deletes the specified TACACS+ server identified by the IPv4 address. By default, no server is configured. |
| **Step 3** | `switch(config)# tacacs-server host 171.71.58.91 port 2` | Configures the TCP port for all TACACS+ requests. |
| | `switch(config)# no tacacs-server host 171.71.58.91 port 2` | Reverts to the factory default of using port 49 for server access. |
| **Step 4** | `switch(config)# tacacs-server host 171.71.58.91 key MyKey` | Configures the TACACS+ server identified by the specified domain name and assigns the secret key. |
| **Step 5** | `switch(config)# tacacs-server host 171.71.58.91 timeout 25` | Configures the timeout period for the switch to wait for a response from the specified server before it declares a timeout failure. |

To configure the TACACS+ server IPv6 address and other options, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# tacacs-server host 2001:0DB8:800:200C::417A`<br>`warning: no key is configured for the host` | Configures the TACACS+ server identified by the specified IPv6 address. |
| | `switch(config)# no tacacs-server host 2001:0DB8:800:200C::417A` | Deletes the specified TACACS+ server identified by the IPv6 address. By default, no server is configured. |
| **Step 3** | `switch(config)# tacacs-server host 2001:0DB8:800:200C::417A port 2` | Configures the TCP port for all TACACS+ requests. |
| | `switch(config)# no tacacs-server host 2001:0DB8:800:200C::417A port 2` | Reverts to the factory default of using port 49 for server access. |
| **Step 4** | `switch(config)# tacacs-server host 2001:0DB8:800:200C::417A key MyKey` | Configures the TACACS+ server identified by the specified domain name and assigns the secret key. |
| **Step 5** | `switch(config)# tacacs-server host 2001:0DB8:800:200C::417A timeout 25` | Configures the timeout period for the switch to wait for a response from the specified server before it declares a timeout failure. |

To configure the TACACS+ server DNS name and other options, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# tacacs-server host host1.cisco.com`<br>`warning: no key is configured for the host` | Configures the TACACS+ server identified by the specified DNS name. |
| | `switch(config)# no tacacs-server host host1.cisco.com` | Deletes the specified TACACS+ server identified by the DNS name. By default, no server is configured. |

| | Command | Purpose |
|---|---|---|
| Step 3 | `switch(config)# tacacs-server host host1.cisco.com port 2` | Configures the TCP port for all TACACS+ requests. |
| | `switch(config)# no tacacs-server host host1.cisco.com port 2` | Reverts to the factory default of using port 49 for server access. |
| Step 4 | `switch(config)# tacacs-server host host1.cisco.com key MyKey` | Configures the TACACS+ server identified by the specified domain name and assigns the secret key. |
| Step 5 | `switch(config)# tacacs-server host host1.cisco.com timeout 25` | Configures the timeout period for the switch to wait for a response from the specified server before it declares a timeout failure. |

# Setting the Global Secret Key

You can configure global values for the secret key for all TACACS+ servers.

**Note** If secret keys are configured for individual servers, those keys override the globally configured key.

**Note** You can use the dollar sign ($) and the percent sign (%) in global secret keys.

To set the secret key for TACACS+ servers, follow these steps:

| | Command | Purpose |
|---|---|---|
| Step 1 | `switch# config t` | Enters configuration mode. |
| Step 2 | `switch(config)# tacacs-server key 7 3sdaA3daKUngd` | Assigns the global secret key (in encrypted format) to access the TACACS+ server. This example specifies **7** to indicate the encrypted format being used. If this global key and the individual server keys are not configured, clear text messages are sent to the TACACS+ server(s). |
| | `switch(config)# no tacacs-server key oldPword` | Deletes the configured global secret key to access the TACACS+ server and reverts to the factory default of allowing access to all configured servers. |

# Setting the Default TACACS+ Server Timeout Interval and Retransmits

By default, a switch retries a TACACS+ server only once. This number can be configured. The maximum is five retries per server. You can also configure the timeout value for the TACACS+ server.

# Setting the Timeout Value

You can configure a global timeout value between transmissions for all TACACS+ servers.

**Note** If timeout values are configured for individual servers, those values override the globally configured values.

To set the global timeout value for TACACS+ servers, follow these steps:

| | Command | Purpose |
|---|---|---|
| Step 1 | `switch# config t` | Enters configuration mode. |
| Step 2 | `switch(config)# tacacs-server timeout 30` | Configures the global timeout period in seconds for the switch to wait for a response from all TACACS+ servers before the switch declares a timeout failure. The time ranges from 1 to 1440 seconds. |
| | `switch(config)# no tacacs-server timeout 30` | Deletes the configured timeout period and reverts to the factory default of 5 seconds. |

# About TACACS+ Servers

By default, the TACACS+ feature is disabled in all switches in the Cisco MDS 9000 Family. Fabric Manager or Device Manager enables the TACACS+ feature automatically when you configure a TACACS+ server.

If a secret key is not configured for a configured server, a warning message is issued if a global key is not configured. If a server key is not configured, the global key (if configured) is used for that server.

**Note** Prior to Cisco MDS SAN-OS Release 2.1(2), you can use the dollar sign ($) in the key but the key must be enclosed in double quotes, for example “k$”. The percent sign (%) is not allowed. In Cisco MDS SAN-OS Release 2.1(2) and later, you can use the dollar sign ($) without double quotes and the percent sign (%) in global secret keys.

You can configure global values for the secret key for all TACACS+ servers.

**Note** If secret keys are configured for individual servers, those keys override the globally configured key.

# Configuring TACACS+ Server Monitoring Parameters

You can configure parameters for monitoring TACACS+ servers.

This section includes the following topics:



## Configuring the TACACS+ Test Idle Timer

The test idle timer specifies the interval during which a TACACS+ server receives no requests before the MDS switch sends out a test packet.

**Note** The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, periodic TACACS+ server monitoring is not performed.

To configure the idle timer, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# tacacs-server host 10.1.1.1 test idle-time 20` | Configures the test idle time interval value in minutes. The valid range is 1 to 1440 minutes. |
| **Step 3** | `switch(config)# no tacacs-server host 10.1.1.1 test idle-time 20` | Reverts to the default value (0 minutes). |

## Configuring Test Username

You can configure a username and password for periodic TACACS+ server status testing. You do not need to configure the user name and password to monitor TACACS+ servers. You can use the default test username (test) and default password (test).

To configure the optional username and password for periodic TACACS+ server status testing, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# tacacs-server host 10.1.1.1 test username testuser` | Configures the test user (testuser) with the default password (test). The default username is test. |
| | `switch(config)# no tacacs-server host 10.1.1.1 test username testuser` | Removes the test user (testuser). |
| | `switch(config)# tacacs-server host 10.1.1.1 test username testuser password Ur2Gd2BH` | Configures the test user (testuser) and assigns a strong password. |

## Configuring the Dead Timer

The dead timer specifies the interval that the MDS switch waits, after declaring a TACACS+ server is dead, before sending out a test packet to determine if the server is now alive.

**Note** The default dead timer value is 0 minutes. TACACS+ server monitoring is not performed if the dead timer interval is 0 minutes, unless the TACACS+ server is a part of a bigger group with the dead-time interval greater than 0 minutes. (See "Configuring RADIUS Server Monitoring Parameters" section on page 4-79).

**Note** If the dead timer of a dead TACACS+ server expires before it is sent a TACACS+ test message, that server is marked as alive again even if it is still not responding. To avoid this scenario, configure a test user with a shorter idle time than the dead timer time.

To configure the dead timer, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# tacacs-server deadtime 30` | Configures the dead-time interval value in minutes. The valid range is 1 to 1440 minutes. |
| | `switch(config)# no tacacs-server deadtime 30` | Reverts to the default value (0 minutes).<br>**Note** When the dead-time interval is 0 minutes, TACACS+ server monitoring is not performed unless the TACACS+ server is part of a server group and the dead-time interval for the group is greater than 0 minutes. (See the "Configuring RADIUS Server Monitoring Parameters" section on page 4-79). |

# Sending TACACS+ Test Messages for Monitoring

You can manually send test messages to monitor a TACACS+ server.

To send the test message to the TACACS+ server, follow these steps:

| Command | Purpose |
|---|---|
| `switch# test aaa server tacacs+ 10.10.1.1 test` | Sends a test message to a TACACS+ server using the default username (test) and password (test). |
| `switch# test aaa server tacacs+ 10.10.1.1 testuser Ur2Gd2BH` | Sends a test message to a TACACS+ server using a configured test username and password.<br>A configured username and password is optional (see the "Configuring Test Username" section on page 4-95). |

# Password Aging Notification through TACACS+ Server

Password aging notification is initiated when the user authenticates to a Cisco MDS 9000 switch via a TACACS+ account. The user is notified when a password is about to expire or has expired. If the password has expired, user is prompted to change the password.

**Note** As of Cisco MDS SAN-OS Release 3.2(1), only TACACS+ supports password aging notification. If you try to use RADIUS servers by enabling this feature, RADIUSs will generate a SYSLOG message and authentication will fall back to the local database.

Password aging notification facilitates the following:

- Password change—You can change your password by entering a blank password.
- Password aging notification—Notifies password aging. Notification happens only if the AAA server is configured and MSCHAP and MSCHAPv2 is disabled.
- Password change after expiration—Initiates password change after the old password expires. Initiation happens from the AAA server.

**Note** Password aging notification fails if you do not disable MSCHAP and MSCHAPv2 authentication.

To enable the password aging option in the AAA server, enter the following command:

```
aaa authentication login ascii-authentication
```

To determine whether or not password aging notification is enabled or disabled in the AAA server, enter the following command:

```
show aaa authentication login ascii-authentication
```

# About Validating a TACACS+ Server

As of Cisco SAN-OS Release 3.0(1), you can periodically validate a TACACS+ server. The switch sends a test authentication to the server using the test username and test password that you configure. If the server does not respond to the test authentication, then the server is considered nonresponding.

**Note** We recommend that you do not configure the test user on your TACACS+ server for security reasons.

You can configure this option to test the server periodically, or you can run a one-time only test.

## Periodically Validating a TACACS+ Server

To configure the switch to periodically test a TACACS+ server using Fabric Manager, see the "Configuring TACACS+ Server Monitoring Parameters" section on page 4-90.

# About Users Specifying a TACACS+ Server at Login

By default, an MDS switch forwards an authentication request to the first server in the TACACS+ server group. You can configure the switch to allow the user to specify which TACACS+ server to send the authenticate request. If you enable this feature, the user can log in as *username@hostname*, where the *hostname* is the name of a configured TACACS+ server.

**Note** User specified logins are supported only for Telnet sessions

# Allowing Users to Specify a TACACS+ Server at Login

To allow users logging into an MDS switch to select a TACACS+ server for authentication, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# tacacs-server directed-request` | Allows users to specify a TACACS+ server to send the authentication request when logging in. |
| | `switch(config)# no tacacs-server directed-request` | Reverts to sending the authentication request to the first server in the server group (default). |

You can use the **show tacacs-server directed-request** command to display the TACACS+ directed request configuration.

```
switch# show tacacs-server directed-request
disabled
```

# Defining Roles on the Cisco Secure ACS 5.x GUI

Enter the following in the GUI under **Policy Elements**:

*Table 4-3 Role Definitions*

| Attribute | Requirement | Value |
|---|---|---|
| shell:roles | Optional | network-admin |

# Defining Custom Attributes for Roles

Cisco MDS 9000 Family switches use the TACACS+ custom attribute for service shells to configure roles to which a user belongs. TACACS+ attributes are specified in **name=value** format. The attribute name for this custom attribute is **cisco-av-pair**. The following example illustrates how to specify roles using this attribute:

```
cisco-av-pair=shell:roles=”network-admin vsan-admin”
```

You can also configure optional custom attributes to avoid conflicts with non-MDS Cisco switches using the same AAA servers.

```
cisco-av-pair*shell:roles="network-admin vsan-admin"
```

Additional custom attribute shell:roles are also supported:

```
shell:roles="network-admin vsan-admin"
```

or

```
shell:roles*"network-admin vsan-admin”
```

**Note** TACACS+ custom attributes can be defined on an Access Control Server (ACS) for various services (for example, shell). Cisco MDS 9000 Family switches require the TACACS+ custom attribute for the service shell to be used for defining roles.

## Supported TACACS+ Server Parameters

The Cisco NX-OS software currently supports the following parameters for the listed TACACS+ servers:

- TACACS+

```
cisco-av-pair=shell:roles="network-admin"
```

- Cisco ACS TACACS+

```
shell:roles="network-admin"
shell:roles*"network-admin"
cisco-av-pair*shell:roles="network-admin"
cisco-av-pair*shell:roles*"network-admin"
cisco-av-pair=shell:roles*"network-admin"
```

- Open TACACS+

```
cisco-av-pair*shell:roles="network-admin"
cisco-av-pair=shell:roles*"network-admin"
```

# Displaying TACACS+ Server Details

Use the **show aaa** and **show tacacs-server** commands to display information about TACACS+ server configuration in all switches in the Cisco MDS 9000 Family as shown in Examples 4-7 to 4-12.

***Example 4-7 Displays Configured TACACS+ Server Information***

```
switch# show tacacs-server
Global TACACS+ shared secret:***********
timeout value:30
total number of servers:3

following TACACS+ servers are configured:
        171.71.58.91:
                available on port:2
        cisco.com:
                available on port:49
        171.71.22.95:
                available on port:49
                TACACS+ shared secret:*****
```

***Example 4-8 Displays AAA Authentication Information***

```
switch# show aaa authentication
         default: group TacServer local none
         console: local
         iscsi: local
         dhchap: local
```

*Example 4-9 Displays AAA Authentication Login Information*

```
switch# show aaa authentication login error-enable
enabled
```

*Example 4-10 Displays Configured TACACS+ Server Groups*

```
switch# show tacacs-server groups
total number of groups:2

following TACACS+ server groups are configured:
        group TacServer:
                server 171.71.58.91 on port 2
        group TacacsServer1:
                server ServerA on port 49
                server ServerB on port 49:
```

*Example 4-11 Displays All AAA Server Groups*

```
switch# show aaa groups
radius
TacServer
```

*Example 4-12 Displays TACACS+ Server Statistics*

```
switch# show tacacs-server statistics 10.1.2.3
Server is not monitored

Authentication Statistics
        failed transactions: 0
        successful transactions: 0
        requests sent: 0
        requests timed out: 0
        responses with no matching requests: 0
        responses not processed: 0
        responses containing errors: 0

Authorization Statistics
        failed transactions: 0
        sucessfull transactions: 0
        requests sent: 0
        requests timed out: 0
        responses with no matching requests: 0
        responses not processed: 0
        responses containing errors: 0

Accounting Statistics
        failed transactions: 0
        successful transactions: 0
        requests sent: 0
        requests timed out: 0
        responses with no matching requests: 0
        responses not processed: 0
        responses containing errors: 0
```

## Clearing TACACS+ Server Statistics

You can clear all the TACACS+ server statistics using the **clear tacacs-server statistics 10.1.2.3** command.

# Configuring Server Groups

You can specify one or more remote AAA servers to authenticate users using server groups. All members of a group must belong to the same protocol, either RADIUS or TACACS+. The servers are tried in the same order in which you configure them.

The AAA server monitoring feature can mark an AAA server as dead. You can configure a period of time in minutes to elapse before the switch sends requests to a dead AAA server. (See the "AAA Server Monitoring" section on page 4-64).

This section includes the following topics:



## About Configuring Server Groups

You can configure these server groups at any time but they only take effect when you apply them to an AAA service. You configure AAA policies for CLI users or Fabric Manager or Device Manager users.

To configure a RADIUS server group, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# aaa group server radius RadServer`<br>`switch(config-radius)#` | Creates a server group named RadServer and enters the RADIUS server group configuration submode for that group. |
| | `switch(config)# no aaa group server radius RadServer` | Deletes the server group called RadServer from the authentication list. |
| **Step 3** | `switch(config-radius)# server 10.71.58.91` | Configures the RADIUS server at IPv4 address 10.71.58.91 to be tried first within the server group RadServer.<br>**Tip** If the specified RADIUS server is not found, configure it using the **radius-server host** command and retry this command. |
| **Step 4** | `switch(config-radius)# server 2001:0DB8:800:200C::417A` | Configures the RADIUS server at IPv6 address 2001:0DB8:800:200C::417A to be tried first within the server group RadServer. |
| | `switch(config-radius)# no server 2001:0DB8:800:200C::417A` | Removes the RADIUS server at IPv6 address 2001:0DB8:800:200C::417A from the server group RadServer. |
| **Step 5** | `switch(config-radius)# exit` | Returns to configuration mode. |

| | Command | Purpose |
|---|---|---|
| **Step 6** | `switch(config)# aaa group server radius RadiusServer`<br>`switch(config-radius)#` | Creates a server group named RadiusServer and enters the RADIUS server group configuration submode for that group. |
| **Step 7** | `switch(config-radius)# server ServerA` | Configures ServerA to be tried first within the server group called the RadiusServer1.<br>**Tip** If the specified RADIUS server is not found, configure it using the **radius-server host** command and retry this command. |
| **Step 8** | `switch(config-radius)# server ServerB` | Configures ServerB to be tried second within the server group RadiusServer1. |
| **Step 9** | `switch(config-radius)# deadtime 30` | Configures the monitoring dead time to 30 minutes. The range is 0 through 1440.<br>**Note** If the dead-time interval for an individual RADIUS server is greater than 0, that value takes precedence over the value set for the server group. |
| | `switch(config-radius)# no deadtime 30` | Reverts to the default value (0 minutes).<br>**Note** If the dead-time interval for both the RADIUS server group and an individual TACACS+ server in the RADIUS server group is set to 0, the switch does not mark the RADIUS server as dead when it is found to be unresponsive by periodic monitoring. Also, the switch does not perform dead server monitoring for that RADIUS server. (See the "Configuring RADIUS Server Monitoring Parameters" section on page 4-83). |

To verify the configured server group order, use the **show radius-server groups** command:

```
switch# show radius-server groups
total number of groups:2

following RAIDUS server groups are configured:
        group RadServer:
                server 10.71.58.91 on port 2
        group RadiusServer1:
                server ServerA on port 49
                server ServerB on port 49:
```

To configure a TACACS+ server group, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# aaa group server tacacs+ TacacsServer1`<br>`switch(config-tacacs+)#` | Creates a server group named TacacsServer1 and enters the submode for that group. |
| | `switch(config)# no aaa group server tacacs+ TacacsServer1` | Deletes the server group called TacacsServer1 from the authentication list. |

| | Command | Purpose |
|---|---|---|
| **Step 3** | `switch(config-tacacs+)# server ServerA` | Configures ServerA to be tried first within the server group called the TacacsServer1.<br>**Tip** If the specified TACACS+ server is not found, configure it using the **tacacs-server host** command and retry this command. |
| **Step 4** | `switch(config-tacacs+)# server ServerB` | Configures ServerB to be tried second within the server group TacacsServer1. |
| | `switch(config-tacacs+)# no server ServerB` | Deletes ServerB within the TacacsServer1 list of servers. |
| **Step 5** | `switch(config-tacacs+)# deadtime 30` | Configures the monitoring dead time to 30 minutes. The range is 0 through 1440.<br>**Note** If the dead-time interval for an individual TACACS+ server is greater than 0, that value takes precedence over the value set for the server group. |
| | `switch(config-tacacs+)# no deadtime 30` | Reverts to the default value (0 minutes).<br>**Note** If the dead-time interval for both the TACACS+ server group and an individual TACACS+ server in the TACACS+ server group is set to 0, the switch does not mark the TACACS+ server as dead when it is found to be unresponsive by periodic monitoring. Also, the switch does not perform dead server monitoring for that TACACS+ server. (See the Configuring TACACS+ Server Monitoring Parameters, page 4-90). |

**Note** Configuration of a TACACS+ group fails if MSCHPv2 authentication is not disabled.

# About Bypassing a Nonresponsive Server

As of Cisco SAN-OS Release 3.0(1), you can bypass a nonresponsive AAA server within a server group. If the switch detects a nonresponsive server, it will bypass that server when authenticating users. Use this feature to minimize login delays caused by a faulty server. Instead of sending a request to a nonresponsive server and waiting for the authentication request to timeout, the switch sends the authentication request to the next server in the server group. If there are no other responding servers in the server group, the switch continues to attempt authentications against the nonresponsive server.

# AAA Server Distribution

Configuration for RADIUS and TACACS+ AAA on an MDS switch can be distributed using the Cisco Fabric Services (CFS). The distribution is disabled by default (see the *Cisco MDS 9000 Family NX-OS System Management Configuration Guide* and the *Cisco Fabric Manager System Management Configuration Guide*).

After enabling the distribution, the first server or global configuration starts an implicit session. All server configuration commands entered thereafter are stored in a temporary database and applied to all switches in the fabric (including the originating one) when you explicitly commit the database. The various server and global parameters are distributed, except the server and global keys. These keys are unique secrets to a switch and should not be shared with other switches.

**Note** Server group configurations are not distributed.

This section includes the following topics:



**Note** For an MDS switch to participate in AAA server configuration distribution, it must be running Cisco MDS SAN-OS Release 2.0(1b) or later, or Cisco NX-OS Release 4.1(1).

## Enabling AAA Server Distribution

Only switches where distribution is enabled can participate in the distribution activity.

To enable RADIUS server distribution, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# radius distribute` | Enables RADIUS configuration distribution in this switch. |
| | `switch(config)# no radius distribute` | Disables RADIUS configuration distribution in this switch (default). |

To enable TACACS+ server distribution, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# tacacs+ distribute` | Enables TACACS+ configuration distribution in this switch. |
| | `switch(config)# no tacacs+ distribute` | Disables TACACS+ configuration distribution in this switch (default). |

# Starting a Distribution Session on a Switch

A distribution session starts the moment you begin a RADIUS/TACACS+ server or global configuration. For example, the following tasks start an implicit session:

- Specifying the global timeout for RADIUS servers.
- Specifying the global timeout for TACACS+ servers.

**Note** After you issue the first configuration command related to AAA servers, all server and global configurations that are created (including the configuration that caused the distribution session start) are stored in a temporary buffer, not in the running configuration.

# Displaying the Session Status

Once the implicit distribution session has started, you can check the session status from Fabric Manager by expanding **Switches > Security > AAA,** and selecting **RADIUS** or **TACACS+.**

Use the **show radius** command to see the distribution status on the CFS tab.

```
switch# show radius distribution status
distribution : enabled
session ongoing: yes
session owner: admin
session db: exists
merge protocol status: merge activation done


last operation: enable
last operation status: success
```

Once the implicit distribution session has started, you can check the session status using the **show tacacs+ distribution status** command.

```
switch# show tacacs+ distribution status
distribution : enabled
session ongoing: yes
session owner: admin
session db: exists
merge protocol status: merge activation done


last operation: enable
last operation status: success
```

# Displaying the Pending Configuration to be Distributed

To display the RADIUS or TACACS+ global and/or server configuration stored in the temporary buffer use the **show radius pending** command, follow these steps:

```
switch(config)# show radius pending-diff
   +radius-server host testhost1 authentication accounting
   +radius-server host testhost2 authentication accounting
```

To display the TACACS+ global and/or server configuration stored in the temporary buffer, use the **show tacacs+ pending** command.

```
switch(config)# show tacacs+ pending-diff
   +tacacs-server host testhost3
   +tacacs-server host testhost4
```

# Committing the Distribution

The RADIUS or TACACS+ global and/or server configuration stored in the temporary buffer can be applied to the running configuration across all switches in the fabric (including the originating switch).

To commit RADIUS configuration changes, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# radius commit` | Commits the RADIUS configuration changes to the running configuration. |

To commit TACACS+ configuration changes, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# tacacs+ commit` | Commits the TACACS+ configuration changes to the running configuration. |

# Discarding the Distribution Session

Discarding the distribution of a session in progress causes the configuration in the temporary buffer to be dropped. The distribution is not applied.

To discard the RADIUS session in-progress distribution, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# radius abort` | Discards the RADIUS configuration changes to the running configuration. |

To discard the TACACS+ session in-progress distribution, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# tacacs+ abort` | Discards the TACACS+ configuration changes to the running configuration. |

# Clearing Sessions

To clear the ongoing CFS distribution session (if any) and to unlock the fabric for the RADIUS feature, enter the **clear radius session** command from any switch in the fabric.

```
switch# clear radius session
```

To clear the ongoing CFS distribution session (if any) and to unlock the fabric for the TACACS+ feature, enter the **clear tacacs+ session** command from any switch in the fabric.

```
switch# clear tacacs+ session
```

# Merge Guidelines for RADIUS and TACACS+ Configurations

The RADIUS and TACACS+ server and global configuration are merged when two fabrics merge. The merged configuration is applied to CFS distribution-enabled switches.

When merging the fabric, be aware of the following conditions:

- The server groups are not merged.
- The server and global keys are not changed during the merge.
- The merged configuration contains all servers found on all CFS enabled switches.
- The timeout and retransmit parameters of the merged configuration are the largest values found per server and global configuration.

**Note** The test parameter will be distributed through CFS for TACACS+ Daemon only. If the fabric contains only NX-OS Release 5.0 switches, then the test parameters will be distributed. If the fabric contains switches running 5.0 versions and some running NX-OS 4.x release, the test parameters will be not distributed.

**Caution** If there is a conflict between two switches in the server ports configured, the merge fails.

Use the **show radius distribution status** command to view the status of the RADIUS fabric merge as shown in Example 4-13.

***Example 4-13 Displays the RADIUS Fabric Merge Status***

```
switch# show radius distribution status
distribution : enabled
session ongoing: no
session db: does not exist
merge protocol status: merge response received
merge error: conflict: server dmtest2 has auth-port 1812 on this switch and 1999
on remote

last operation: enable
last operation status: success
```

Use the **show tacacs+ distribution status** command to view the status of the TACACS+ fabric merge as shown in Example 4-14.

***Example 4-14 Displays the TACACS+ Fabric Merge Status***

```
switch# show tacacs+ distribution status
distribution : enabled
session ongoing: no
session db: does not exist
merge protocol status: merge activation done
```

```
last operation: enable
last operation status: success
```

# CHAP Authentication

CHAP (Challenge Handshake Authentication Protocol) is a challenge-response authentication protocol that uses the industry-standard Message Digest 5 (MD5) hashing scheme to encrypt the response. CHAP is used by various vendors of network access servers and clients. A server running routing and Remote Access supports CHAP so that remote access clients that require CHAP are authenticated. CHAP is supported as an authentication method in this release.

## Enabling CHAP Authentication

To enable CHAP authentication, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# aaa authentication login chap enable` | Enables CHAP login authentication. |
| | `switch# no aaa authentication login chap enable` | Disables CHAP login authentication. |

You can use the **show aaa authentication login chap** command to display the CHAP authentication configuration.

```
switch# show aaa authentication login chap
chap is disabled
```

# MSCHAP Authentication

Microsoft Challenge Handshake Authentication Protocol (MSCHAP) is the Microsoft version of CHAP.

Cisco MDS 9000 Family switches allow user logins to perform remote authentication using different versions of MSCHAP. MSCHAP is used for authentication on a RADIUS or TACACS+ server, while MSCHAPv2 is used for authentication on a RADIUS server.

## About Enabling MSCHAP

By default, the switch uses Password Authentication Protocol (PAP) authentication between the switch and the remote server. If you enable MSCHAP, you need to configure your RADIUS server to recognize the MSCHAP vendor-specific attributes. See the "About Vendor-Specific Attributes" section on page 4-87. Table 4-4 shows the RADIUS vendor-specific attributes required for MSCHAP.

*Table 4-4 MSCHAP RADIUS Vendor-Specific Attributes*

| Vendor-ID Number | Vendor-Type Number | Vendor-Specific Attribute | Description |
|---|---|---|---|
| 311 | 11 | MSCHAP-Challenge | Contains the challenge sent by an AAA server to an MSCHAP user. It can be used in both Access-Request and Access-Challenge packets. |
| 211 | 11 | MSCHAP-Response | Contains the response value provided by an MS-CHAP user in response to the challenge. It is only used in Access-Request packets. |

# Enabling MSCHAP Authentication

To enable MSCHAP authentication, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# aaa authentication login mschap enable` | Enables MSCHAP login authentication. |
| **Step 3** | `switch# no aaa authentication login mschap enable` | Disables MSCHAP login authentication. |

To enable MSCHAPv2 authentication, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# aaa authentication login mschapv2 enable` | Enables MSCHAPv2 login authentication. |
| **Step 3** | `switch# no aaa authentication login mschapv2 enable` | Disables MSCHAPv2 login authentication. |

**Note** Password aging, MSCHAPv2 and MSCHAP authentication can fail if one of these authentication is not disabled.

**Note** A warning message is issued when you execute a command to enable MSCHAPv2 authentication on the TACACS+ server, and the configuration fails.

You can use the **show aaa authentication login mschap** command to display the MSCHAP authentication configuration.

```
switch# show aaa authentication login mschap
mschap is disabled
```

You can use the show aaa authentication login mschapv2 command to display the MSCHAPv2 authentication configuration.

```
switch# show aaa authentication login mschapv2
mschapv2 is enabled
```

# Local AAA Services

The system maintains the username and password locally and stores the password information in encrypted form. You are authenticated based on the locally stored user information.

Use the **username** command to configure local users and their roles.

Use the **show accounting log** command to view the local accounting log as shown in Example 4-15.

***Example 4-15 Displays the Accounting Log Information***

```
switch# show accounting log

Thu Dec 10 06:19:21 2009:type=update:id=console0:user=root:cmd=enabled telnet
Thu Dec 10 06:19:21 2009:type=update:id=console0:user=root:cmd=configure terminal ;
feature telnet (SUCCESS)
Thu Dec 10 06:19:35 2009:type=start:id=171.69.16.56@pts/1:user=admin:cmd=
Thu Dec 10 06:20:16 2009:type=stop:id=171.69.16.56@pts/1:user=admin:cmd=shell te
rminated gracefully
Thu Dec 10 06:20:20 2009:type=stop:id=console0:user=root:cmd=shell terminated gr
acefully
Thu Dec 10 06:29:37 2009:type=start:id=72.163.177.168@pts/1:user=admin:cmd=
Thu Dec 10 06:29:42 2009:type=update:id=72.163.177.168@pts/1:user=admin:cmd=pwd
(SUCCESS)
Thu Dec 10 06:32:49 2009:type=start:id=72.163.190.8@pts/2:user=admin:cmd=
```

# Disabling AAA Authentication

You can turn off password verification using the **none** option. If you configure this option, users can log in without giving a valid password. But the user should at least exist locally on the Cisco MDS 9000 Family switch.

**Caution** Use this option cautiously. If configured, any user can access the switch at any time.

Use the **none** option in the **aaa authentication login** command to disable password verification.

A user created by entering the **username** command will exist locally on the Cisco MDS 9000 Family switch.

# Displaying AAA Authentication

The **show aaa authentication** command displays the configured authentication methods as shown in Example 4-16.

***Example 4-16 Displays Authentication Information***

```
switch# show aaa authentication

    No AAA Authentication
    default: group TacServer local none
```

```
console: local none
iscsi: local
dhchap: local
```

# Configuring Accounting Services

Accounting refers to the log information that is kept for each management session in a switch. This information may be used to generate reports for troubleshooting and auditing purposes. Accounting can be implemented locally or remotely (using RADIUS). The default maximum size of the accounting log is 250,000 bytes and cannot be changed.

**Tip** The Cisco MDS 9000 Family switch uses interim-update RADIUS accounting-request packets to communicate accounting log information to the RADIUS server. The RADIUS server must be appropriately configured to log the information communicated in these packets. Several servers typically have log update/watchdog packets flags in the AAA client configuration. Turn on this flag to ensure proper RADIUS accounting.

**Note** Configuration operations are automatically recorded in the accounting log if they are performed in configuration mode. Additionally, important system events (for example, configuration save and system switchover) are also recorded in the accounting log.

# Displaying Accounting Configuration

To display configured accounting information use **show accounting** command. See Examples 4-17 to 4-19. To specify the size of the local accounting log to be displayed, use the **show accounting log** command. By default approximately 250 KB of the accounting log is displayed.

***Example 4-17 Displays Two Samples of Configured Accounting Parameters***

```
switch# show accounting config
show aaa accounting
               default: local

switch# show aaa accounting
               default: group rad1
```

***Example 4-18 Displays 60,000 Bytes of the Accounting Log***

```
switch# show accounting log 60000
Fri Jan 16 15:28:21 1981:stop:snmp_348506901_64.104.131.208:admin:
Fri Jan 16 21:17:04 1981:start:/dev/pts/0_348527824:admin:
Fri Jan 16 21:35:45 1981:update:/dev/pts/0_348527824:admin:updated RADIUS parameters for
group:Group1
Fri Jan 16 21:35:51 1981:update:/dev/pts/0_348527824:admin:updated RADIUS parameters for
group:Group1
Fri Jan 16 21:35:51 1981:update:/dev/pts/0_348527824:admin:updated RADIUS parameters for
group:Group5
Fri Jan 16 21:35:55 1981:update:/dev/pts/0_348527824:admin:updated RADIUS parameters for
group:Group5
```

```
Fri Jan 16 21:35:55 1981:update:/dev/pts/0_348527824:admin:updated RADIUS parameters for
group:Group3
Fri Jan 16 21:58:17 1981:start:snmp_348530297_171.71.150.105:admin:
...
```

*Example 4-19 Displays the Entire Log File*

```
switch# show accounting log
Fri Jan 16 15:28:21 1981:stop:snmp_348506901_64.104.131.208:admin:
Fri Jan 16 21:17:04 1981:start:/dev/pts/0_348527824:admin:
Fri Jan 16 21:35:45 1981:update:/dev/pts/0_348527824:admin:updated RADIUS parameters for
group:Group1
Fri Jan 16 21:35:51 1981:update:/dev/pts/0_348527824:admin:updated RADIUS parameters for
group:Group1
Fri Jan 16 21:35:51 1981:update:/dev/pts/0_348527824:admin:updated RADIUS parameters for
group:Group5
Fri Jan 16 21:35:55 1981:update:/dev/pts/0_348527824:admin:updated RADIUS parameters for
group:Group5
Fri Jan 16 21:35:55 1981:update:/dev/pts/0_348527824:admin:updated RADIUS parameters for
group:Group3
Fri Jan 16 21:58:17 1981:start:snmp_348530297_171.71.150.105:admin:
Fri Jan 16 21:58:17 1981:stop:snmp_348530297_171.71.150.105:admin:
Fri Jan 16 21:58:18 1981:start:snmp_348530298_171.71.150.105:admin:
Fri Jan 16 21:58:18 1981:stop:snmp_348530298_171.71.150.105:admin:
...
Fri Jan 16 23:37:02 1981:update:/dev/pts/0_348527824:admin:updated RADIUS parameters for
group:Group3
Fri Jan 16 23:37:26 1981:update:/dev/pts/0_348527824:admin:updated TACACS+ parameters for
group:TacacsServer1
Fri Jan 16 23:45:19 1981:update:/dev/pts/0_348527824:admin:updated TACACS+ parameters for
group:TacacsServer1
Fri Jan 16 23:45:19 1981:update:/dev/pts/0_348527824:admin:updated RADIUS parameters for
group:Group1
...
Fri Jan 16 23:53:51 1981:update:/dev/pts/0_348527824:admin:updated RADIUS parameters for
server:Server3
Fri Jan 16 23:54:00 1981:update:/dev/pts/0_348527824:admin:updated RADIUS parameters for
server:Server5
Fri Jan 16 23:54:22 1981:update:/dev/pts/0_348527824:admin:updated TACACS+ parameters for
server:ServerA
Fri Jan 16 23:54:25 1981:update:/dev/pts/0_348527824:admin:updated TACACS+ parameters for
server:ServerB
Fri Jan 16 23:55:03 1981:update:/dev/pts/0_348527824:admin:updated RADIUS parameters for
group:Group1
...
Sat Jan 17 00:01:41 1981:start:snmp_348537701_171.71.58.100:admin:
Sat Jan 17 00:01:41 1981:stop:snmp_348537701_171.71.58.100:admin:
Sat Jan 17 00:01:42 1981:start:snmp_348537702_171.71.58.100:admin:
Sat Jan 17 00:01:42 1981:stop:snmp_348537702_171.71.58.100:admin:
...
```

# Clearing Accounting Logs

To clear out the contents of the current log, use the **clear accounting log** command.

```
switch# clear accounting log
```

# Configuring Cisco Access Control Servers

The Cisco Access Control Server (ACS) uses TACACS+ and RADIUS protocols to provide AAA services that ensure a secure environment.When using the AAA server, user management is normally done using Cisco ACS. Figure 4-3, Figure 4-4, Figure 4-5, and Figure 4-6 display ACS server user setup configurations for network-admin roles and multiple roles using either RADIUS or TACACS+.

*Figure 4-3 Configuring the network-admin Role When Using RADIUS*

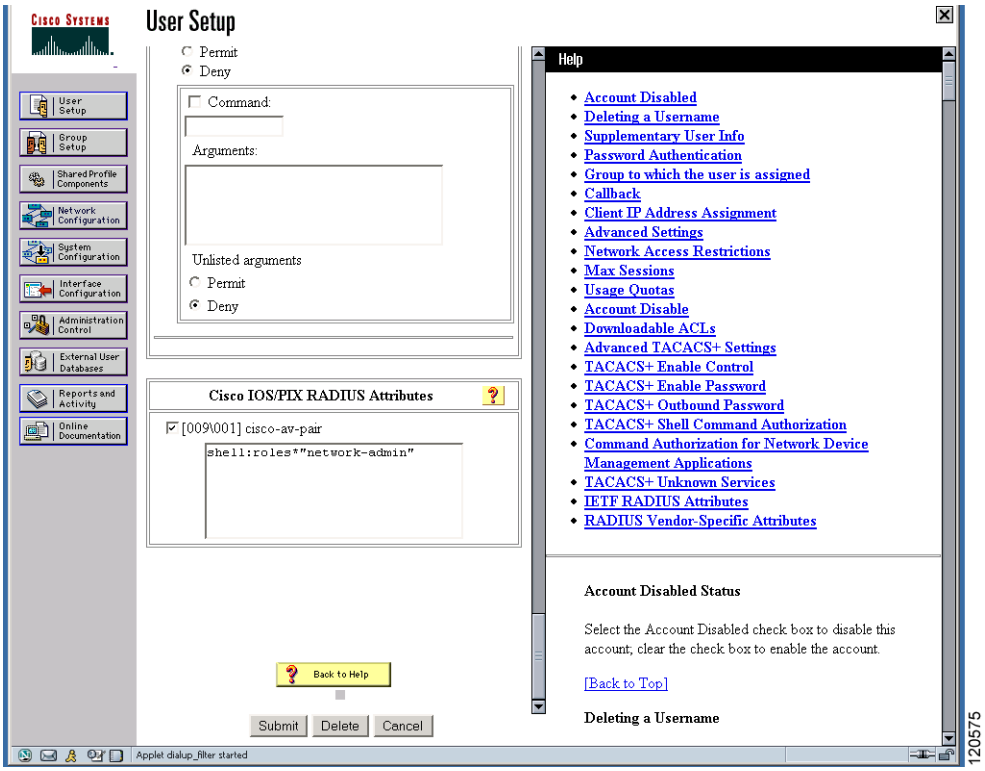


*Figure 4-4 Configuring Multiple Roles with SNMPv3 Attributes When Using RADIUS*

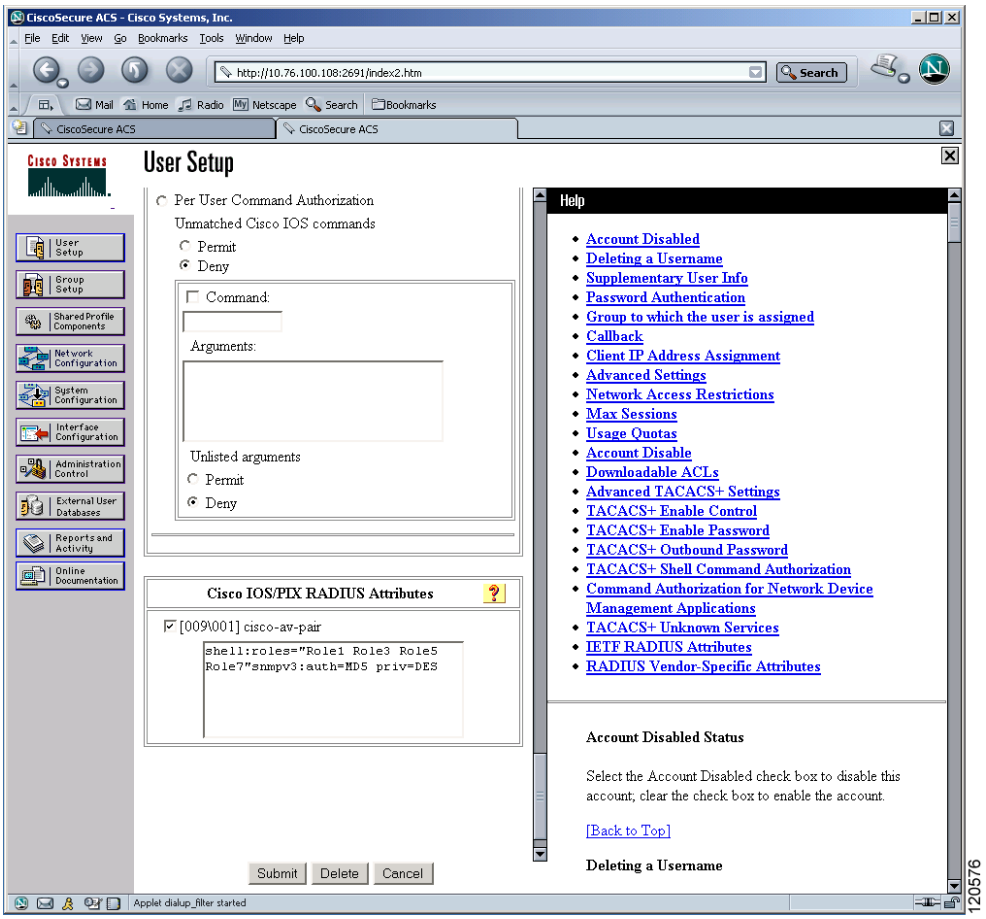


*Figure 4-5 Configuring the network-admin Role with SNMPv3 Attributes When Using TACACS+*

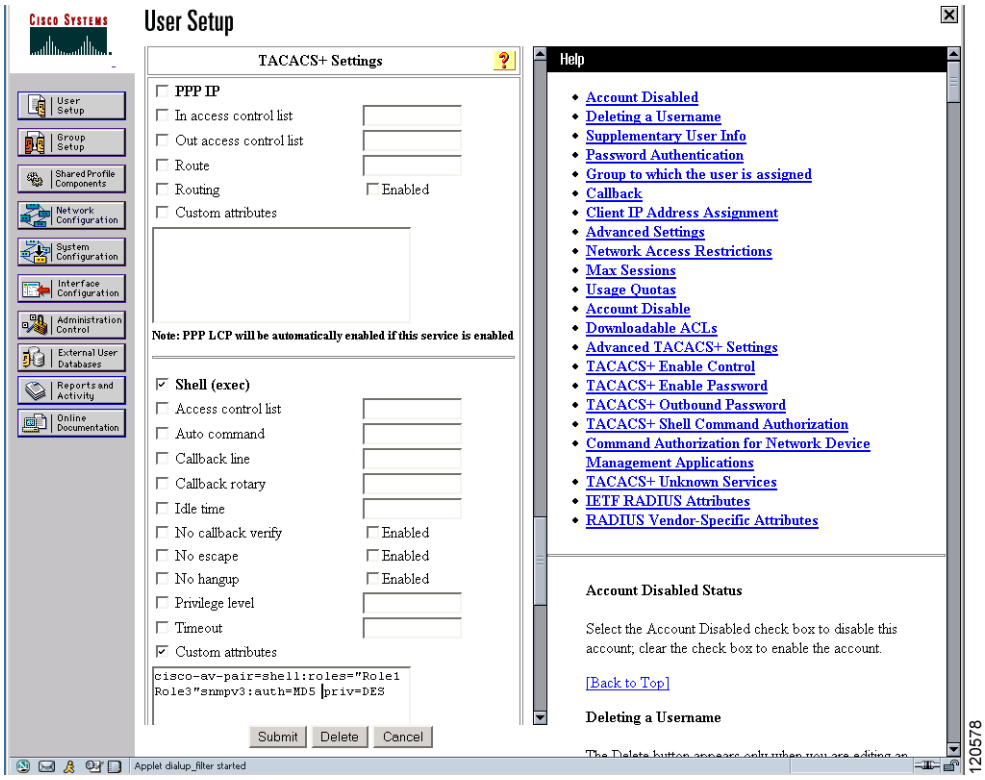
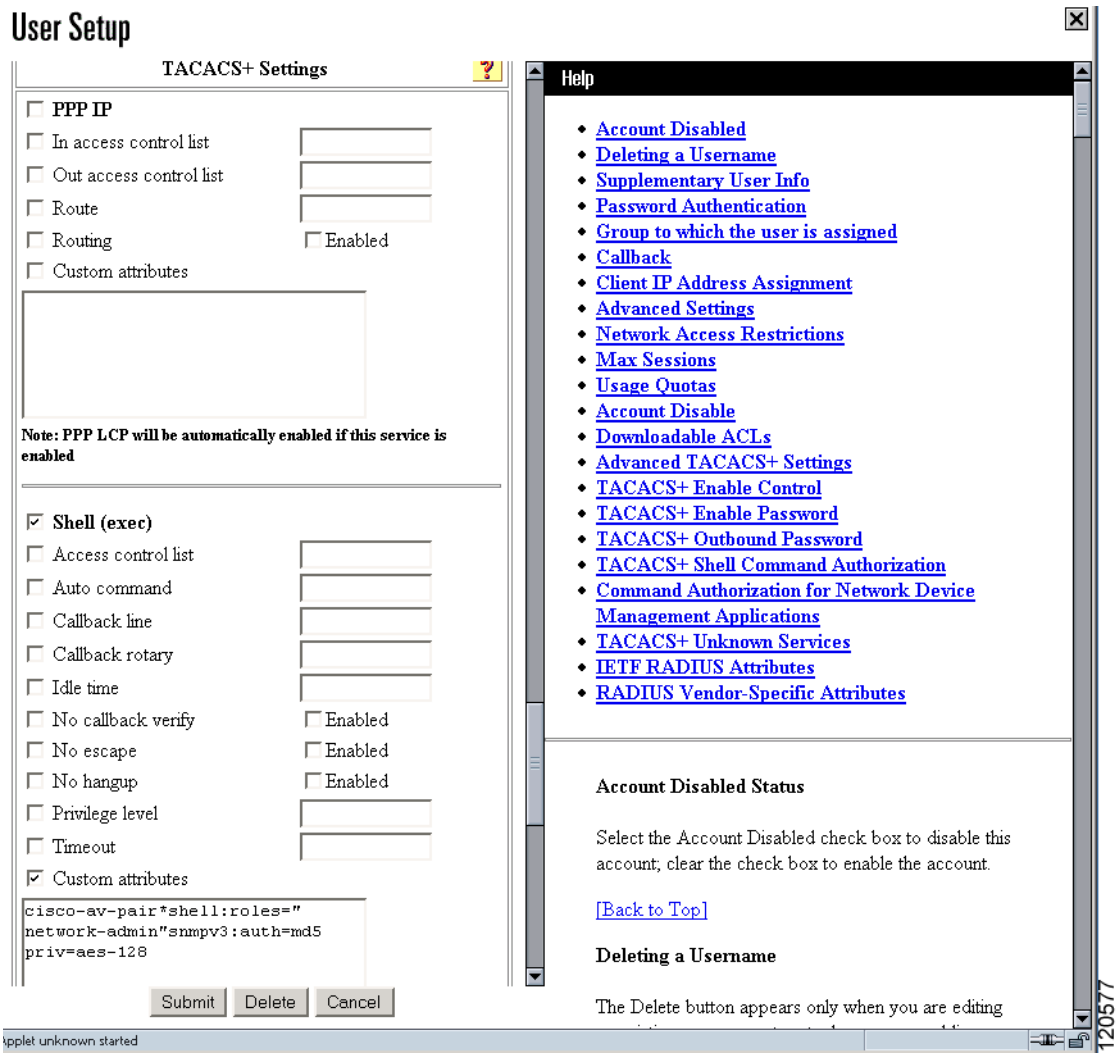


*Figure 4-6 Configuring Multiple Roles with SNMPv3 Attributes When Using TACACS+*




# Default Settings

Table 4-5 lists the default settings for all switch security features in any switch.

*Table 4-5 Default Switch Security Settings*

| Parameters | Default |
|---|---|
| Roles in Cisco MDS switches | Network operator (network-operator) |
| AAA configuration services | Local |
| Authentication port | 1812 |
| Accounting port | 1813 |
| Preshared key communication | Clear text |

*Table 4-5 Default Switch Security Settings (continued)*

| Parameters | Default |
|---|---|
| RADIUS server timeout | 1 (one) second |
| RADIUS server retries | Once |
| Authorization | Disabled |
| aaa user default role | enabled |
| RADIUS server directed requests | Disabled |
| TACACS+ | Disabled |
| TACACS+ servers | None configured |
| TACACS+ server timeout | 5 seconds |
| TACACS+ server directed requests | Disabled |
| AAA server distribution | Disabled |
| Accounting log size | 250 KB |


# Configuring IPv4 and IPv6 Access Control Lists

Cisco MDS 9000 Family switches can route IP version 4 (IPv4) traffic between Ethernet and Fibre Channel interfaces. The IP static routing feature routes traffic between VSANs. To do so, each VSAN must be in a different IPv4 subnetwork. Each Cisco MDS 9000 Family switch provides the following services for network management systems (NMS):

- IP forwarding on the out-of-band Ethernet interface (mgmt0) on the front panel of the supervisor modules.
- IP forwarding on the in-band Fibre Channel interface using the IP over Fibre Channel (IPFC) function—IPFC specifies how IP frames can be transported over Fibre Channel using encapsulation techniques. IP frames are encapsulated into Fibre Channel frames so NMS information can cross the Fibre Channel network without using an overlay Ethernet network.
- IP routing (default routing and static routing)—If your configuration does not need an external router, you can configure a default route using static routing.

Switches are compliant with RFC 2338 standards for Virtual Router Redundancy Protocol (VRRP) features. VRRP is a restartable application that provides a redundant, alternate path to the gateway switch.

IPv4 Access Control Lists (IPv4-ACLs and IPv6-ACLs) provide basic network security to all switches in the Cisco MDS 9000 Family. IPv4-ACLs and IPv6-ACLs restrict IP-related traffic based on the configured IP filters. A filter contains the rules to match an IP packet, and if the packet matches, the rule also stipulates if the packet should be permitted or denied.

Each switch in the Cisco MDS 9000 Family can have a maximum total of 128 IPv4-ACLs or 128 IPv6-ACLs and each IPv4-ACL or IPv6-ACL can have a maximum of 256 filters.

This chapter includes the following sections:

# About IPv4 and IPv6 Access Control Lists

Cisco MDS 9000 Family switches can route IP version 4 (IPv4) traffic between Ethernet and Fibre Channel interfaces. The IP static routing feature routes traffic between VSANs. To do so, each VSAN must be in a different IPv4 subnetwork. Each Cisco MDS 9000 Family switch provides the following services for network management systems (NMS):

- IP forwarding on the out-of-band Ethernet interface (mgmt0) on the front panel of the supervisor modules.
- IP forwarding on the in-band Fibre Channel interface using the IP over Fibre Channel (IPFC) function—IPFC specifies how IP frames can be transported over Fibre Channel using encapsulation techniques. IP frames are encapsulated into Fibre Channel frames so NMS information can cross the Fibre Channel network without using an overlay Ethernet network.
- IP routing (default routing and static routing)—If your configuration does not need an external router, you can configure a default route using static routing.

IPv4 Access Control Lists (IPv4-ACLs and IPv6-ACLs) provide basic network security to all switches in the Cisco MDS 9000 Family. IPv4-ACLs and IPv6-ACLs restrict IP-related traffic based on the configured IP filters. A filter contains the rules to match an IP packet, and if the packet matches, the rule also stipulates if the packet should be permitted or denied.

Each switch in the Cisco MDS 9000 Family can have a maximum total of 128 IPv4-ACLs or 128 IPv6-ACLs and each IPv4-ACL or IPv6-ACL can have a maximum of 256 filters.

# IPv4-ACL and IPv6-ACL Configuration Guidelines

Follow these guidelines when configuring IPv4-ACLs or IPv6-ACLs in any switch or director in the Cisco MDS 9000 Family:

- You can apply IPv4-ACLs or IPv6-ACLs to VSAN interfaces, the management interface, Gigabit Ethernet interfaces on IPS modules and MPS-14/2 modules, and Ethernet PortChannel interfaces.

**Caution** If IPv4-ACLs or IPv6-ACLs are already configured in a Gigabit Ethernet interface, you cannot add this interface to an Ethernet PortChannel group. Do not apply IPv4-ACLs or IPv6-ACLs to only one member of a PortChannel group. Apply IPv4-ACLs or IPv6-ACLs to the entire channel group.

- Configure the order of conditions accurately. As the IPv4-ACL or the IPv6-ACL filters are sequentially applied to the IP flows, only the first match determines the action taken. Subsequent matches are not considered. Be sure to configure the most important condition first. If no conditions match, the software drops the packet.
- Configure explicit deny on the IP Storage Gigabit Ethernet ports to apply IP ACLs because implicit deny does not take effect on these ports.

# About Filter Contents

An IP filter contains rules for matching an IP packet based on the protocol, address, port, ICMP type, and type of service (TS).

This section includes the following topics:

# Protocol Information

The protocol information is required in each filter. It identifies the name or number of an IP protocol. You can specify the IP protocol in one of two ways:

- Specify an integer ranging from 0 to 255. This number represents the IP protocol.
- Specify the name of a protocol including, but not restricted to, Internet Protocol (IP), Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Internet Control Message Protocol (ICMP).

**Note** When configuring IPv4-ACLs or IPv6-ACLs on Gigabit Ethernet interfaces, only use the TCP or ICMP options.

# Address Information

The address information is required in each filter. It identifies the following details:

- Source—The address of the network or host from which the packet is being sent.
- Source-wildcard—The wildcard bits applied to the source.
- Destination—The number of the network or host to which the packet is being sent.
- Destination-wildcard—The wildcard bits applied to the destination.

Specify the source and source-wildcard or the destination and destination-wildcard in one of two ways:

- Using the 32-bit quantity in four-part, dotted decimal format (10.1.1.2/0.0.0.0 is the same as host 10.1.1.2).
  - Each wildcard bit set to zero indicates that the corresponding bit position in the packet's IPv4 address must exactly match the bit value in the corresponding bit position in the source.
  - Each wildcard bit set to one indicates that both a zero bit and a one bit in the corresponding position of the packet's IPv4 or IPv6 address will be considered a match to this access list entry. Place ones in the bit positions you want to ignore. For example, 0.0.255.255 requires an exact match of only the first 16 bits of the source. Wildcard bits set to one do not need to be contiguous in the source-wildcard. For example, a source-wildcard of 0.255.0.64 would be valid.
- Using the **any** option as an abbreviation for a source and source-wildcard or destination and destination-wildcard (0.0.0.0/255.255.255.255)

# Port Information

The port information is optional. To compare the source and destination ports, use the **eq** (equal) option, the **gt** (greater than) option, the **lt** (less than) option, or the **range** (range of ports) option. You can specify the port information in one of two ways:

- Specify the number of the port. Port numbers range from 0 to 65535. Table 5-1 displays the port numbers recognized by the Cisco NX-OS software for associated TCP and UDP ports.
- Specify the name of a TCP or UDP port as follows:
  - TCP port names can only be used when filtering TCP.
  - UDP port names can only be used when filtering UDP.

*Table 5-1 TCP and UDP Port Numbers*

| Protocol | Port | Number |
|---|---|---|
| UDP | dns | 53 |
| | tftp | 69 |
| | ntp | 123 |
| | radius accounting | 1646 or 1813 |
| | radius authentication | 1645 or 1812 |
| | snmp | 161 |
| | snmp-trap | 162 |
| | syslog | 514 |
| TCP[1] | ftp | 20 |
| | ftp-data | 21 |
| | ssh | 22 |
| | telnet | 23 |
| | smtp | 25 |
| | tasacs-ds | 65 |
| | www | 80 |
| | sftp | 115 |
| | http | 143 |
| | wbem-http | 5988 |
| | wbem-https | 5989 |

1. If the TCP connection is already established, use the **established** option to find matches. A match occurs if the TCP datagram has the ACK, FIN, PSH, RST, or URG control bit set.

# ICMP Information

IP packets can be filtered based on the following optional ICMP conditions:

- icmp-type—The ICMP message type is a number from 0 to 255.
- icmp-code—The ICMP message code is a number from 0 to 255.

Table 5-2 displays the value for each ICMP type.

*Table 5-2 ICMP Type Value*

| ICMP Type[1] | Code |
|---|---|
| echo | 8 |
| echo-reply | 0 |
| destination unreachable | 3 |
| traceroute | 30 |
| time exceeded | 11 |

1. ICMP redirect packets are always rejected.

## ToS Information

IP packets can be filtered based on the following optional ToS conditions:

- ToS level—The level is specified by a number from 0 to 15.
- ToS name—The name can be max-reliability, max-throughput, min-delay, min-monetary-cost, and normal.

# Creating IPv4-ACLs or IPv6-ACLs

Traffic coming into the switch is compared to IPv4-ACL or IPv6-ACL filters based on the order that the filters occur in the switch. New filters are added to the end of the IPv4-ACL or the IPv6-ACL. The switch keeps looking until it has a match. If no matches are found when the switch reaches the end of the filter, the traffic is denied. For this reason, you should have the frequently hit filters at the top of the filter. There is an *implied deny* for traffic that is not permitted. A single-entry IPv4-ACL or IPv6-ACL with only one deny entry has the effect of denying all traffic.

To configure an IPv4-ACL or an IPv6-ACL, follow these steps:

**Step 1** Create an IPv4-ACL or an IPv6-ACL by specifying a filter name and one or more access condition(s). Filters require the source and destination address to match a condition. Use optional keywords to configure finer granularity.

**Note** The filter entries are executed in sequential order. You can only add the entries to the end of the list. Take care to add the entries in the correct order.

**Step 2** Apply the access filter to specified interfaces.

# Creating IPv4-ACLs or IPv6-ACLs

To create an IPv4-ACL, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# ip access-list List1 permit ip any any` | Configures an IPv4-ACL called List1 and permits IP traffic from any source address to any destination address. |
| | `switch(config)# no ip access-list List1 permit ip any any` | Removes the IPv4-ACL called List1. |
| **Step 3** | `switch(config)# ip access-list List1 deny tcp any any` | Updates List1 to deny TCP traffic from any source address to any destination address. |

To create an IPv6-ACL, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# ipv6 access-list List1`<br>`switch(config-ipv6-acl)#` | Configures an IPv6-ACL called List1 and enters IPv6-ACL configuration submode. |
| | `switch(config)# no ipv6 access-list List1` | Removes the IPv6-ACL called List1 and all its entries. |
| **Step 3** | `switch(config-ipv6-acl)# permit ipv6 any any` | Adds an entry permitting IPv6 traffic from any source address to any destination address. |
| | `switch(config-ipv6-acl)# no permit ipv6 any any` | Removes an entry from the IPv6-ACL. |
| | `switch(config-ipv6-acl)# deny tcp any any` | Adds an entry to deny TCP traffic from any source address to any destination address. |

To define an IPv4-ACL that restricts management access, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# ip access-list restrict_mgmt permit ip 10.67.16.0 0.0.0.255 any` | Defines an entry in an IPv4-ACL named restrict_mgmt allowing all addresses in the 10.67.16.0/24 subnet. |
| **Step 3** | `switch(config)# ip access-list restrict_mgmt permit icmp any any eq 8` | Adds an entry to an IPv4-ACL named restrict_mgmt to allow any device to ping the MDS (icmp type 8). |
| **Step 4** | `switch(config)# ip access-list restrict_mgmt deny ip any any` | Explicitly blocks all other access to an access-list named restrict_mgmt. |

To define an IPv6-ACL that restricts management access, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# ip access-list RestrictMgmt`<br>`switch(config-ipv6-acl)#` | Configures an IPv6-ACL called RestrictMgmt and enters IPv6-ACL configuration submode. |
| **Step 3** | `switch(config)# permit ipv6 2001:0DB8:800:200C::/64 any` | Defines an entry allowing all addresses in the 2001:0DB8:800:200C::/64 prefix. |
| **Step 4** | `switch(config)# permit icmp any any eq 8` | Adds an entry to allow any device to ping the MDS (ICMP type 8). |
| **Step 5** | `switch(config)# deny ipv6 any any` | Explicitly blocks all other IPv6 access. |

To use the operand and port options for an IPv4-ACL, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# ip access-list List2 deny tcp 1.2.3.0 0.0.0.255 eq port 5 any` | Denies TCP traffic from 1.2.3.0 through source port 5 to any destination. |

To use the operand and port options for an IPv6-ACL, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# ip access-list List2 deny tcp 2001:0DB8:800:200C::/64 eq port 5 any` | Denies TCP traffic from 2001:0DB8:800:200C::/64 through source port 5 to any destination. |

# Adding IP Filters to an Existing IPv4-ACL or IPv6-ACL

After you create an IPv4-ACL or an IPv6-ACL, you can add subsequent IP filters at the end of the IPv4-ACL or the IPv6-ACL. You cannot insert filters in the middle of an IPv4-ACL or an IPv6-ACL. Each configured entry is automatically added to the end of a IPv4-ACL or a IPv6-ACL.

To add entries to an existing IPv4-ACL, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# ip access-list List1 permit tcp 10.1.1.2 0.0.0.0 172.16.1.1 0.0.0.0 eq port telnet` | Permits TCP for Telnet traffic. |
| **Step 3** | `switch(config)# ip access-list List1 permit tcp 10.1.1.2 0.0.0.0 172.16.1.1 0.0.0.0 eq port http` | Permits TCP for HTTP traffic. |
| **Step 4** | `switch(config)# ip access-list List1 permit udp 10.1.1.2 0.0.0.0 172.16.1.1 0.0.0.0` | Permits UDP for all traffic. |

To add entries to an existing IPv6-ACL, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# ipv6 access-list List2`<br>`switch(config-ipv6-acl)#` | Configures an IPv6-ACL and enters IPv6-ACL configuration submode. |
| **Step 3** | `switch(config-ipv6-acl)# permit ip 2001:0DB8:800:200C::/64 2001:0DB8:800:2010::/64 eq 23` | Permits TCP for Telnet traffic. |
| **Step 4** | `switch(config-ipv6-acl)# permit tcp 2001:0DB8:800:200C::/64 2001:0DB8:800:2010::/64 eq 143` | Permits TCP for HTTP traffic. |
| **Step 5** | `switch(config-ipv6-acl)# permit udp 2001:0DB8:800:200C::/64 2001:0DB8:800:2010::/64` | Permits UDP for all traffic. |

# Removing IP Filters from an Existing IPv4-ACL or IPv6-ACL

To remove configured entries from an IPv4-ACL, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# no ip access-list List2 deny tcp 1.2.3.0 0.0.0.255 eq port 5 any` | Removes this entry from the IPv4-ACL (List2). |
| | `switch(config)# no ip access-list x3 deny ip any any` | Removes this entry from the IPv4-ACL (x3). |
| | `switch(config)# no ip access-list x3 permit ip any any` | Removes this entry from the IPv4-ACL (x3). |

To remove configured entries from an IPv6-ACL, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# ipv6 access-list List3`<br>`switch(config-ipv6-acl)#` | Configures an IPv6-ACL and enters IPv6-ACL configuration submode. |
| **Step 3** | `switch(config-ipv6-acl)# no deny tcp 2001:0DB8:800:2010::/64 eq port 5 any` | Removes the TCP entry from the IPv6-ACL. |
| **Step 4** | `switch(config-ipv6-acl)# no deny ip any any` | Removes the IP entry from the IPv6-ACL. |

# Verifying the IPv4-ACL or IPv6-ACL Configuration

Use the **show ip access-list** command to view the contents of configured IPv4-ACLs. An IPv4-ACL can have one or more filters. (See Example 5-1).

*Example 5-1 Displays Filters Configured for an IPv4-ACL*

```
switch# show ip access-list abc
ip access-list abc permit tcp any any (0 matches)
ip access-list abc permit udp any any (0 matches)
ip access-list abc permit icmp any any (0 matches)
ip access-list abc permit ip 10.1.1.0 0.0.0.255 (2 matches)
ip access-list abc permit ip 10.3.70.0 0.0.0.255 (7 matches)
```

Use the **show ipv6 access-list** command to view the contents of configured access filters. Each access filter can have several conditions. (See Example 5-2 and Example 5-3).

*Example 5-2 Displays Configured IPv6-ACLs*

```
switch# show ipv6 access-list
switch# show ipv6 access-list

IPv6 access list copp-system-acl-bgp6
        10 permit tcp any gt 1024 any eq bgp
        20 permit tcp any eq bgp any gt 1024
IPv6 access list copp-system-acl-icmp6
        10 permit icmp any any echo-request
        20 permit icmp any any echo-reply
IPv6 access list copp-system-acl-icmp6-msgs
        10 permit icmp any any router-advertisement
        20 permit icmp any any router-solicitation
        30 permit icmp any any nd-na
        40 permit icmp any any nd-ns
        50 permit icmp any any mld-query
        60 permit icmp any any mld-report
        70 permit icmp any any mld-reduction
IPv6 access list copp-system-acl-ntp6
        10 permit udp any any eq ntp
        20 permit udp any eq ntp any
IPv6 access list copp-system-acl-ospf6
        10 permit 89 any any
IPv6 access list copp-system-acl-pim6
        10 permit 103 any ff02::d/128
        20 permit udp any any eq pim-auto-rp
IPv6 access list copp-system-acl-radius6
```

*Example 5-3 Displays a Summary of the Specified IPv6-ACL*

```
switch# show ipv6 access-list abc
```

# Reading the IP-ACL Log Dump

Use the **log-deny** option at the end of a filter condition to log information about packets that match dropped entries. The log output displays the ACL number, permit or deny status, and port information.

**Note** To capture these messages in a logging destination, you must configure severity level 7 for the kernel and ipacl facilities and severity level 7 for the logging destination: logfile, monitor. For example:

```
switch# config t
```

```
switch(config)# logging level kernel 7
switch(config)# logging level ipacl 7
switch(config)# logging logfile message 7
```

For the input ACL, the log displays the raw MAC information. The keyword "MAC=" does not refer to showing an Ethernet MAC frame with MAC address information. It refers to the Layer 2 MAC-layer information dumped to the log. For the output ACL, the raw Layer 2 information is not logged.

The following example is an input ACL log dump:

```
Jul 17 20:38:44 excal-2
%KERN-7-SYSTEM_MSG:
%IPACL-7-DENY:IN=vsan1 OUT=
MAC=10:00:00:05:30:00:47:df:10:00:00:05:30:00:8a:1f:aa:aa:03:00:00:00:08:00:45:00:00:54:00
:00:40:00:40:01:0e:86:0b:0b:0b:0c:0b:0b:0b:02:08:00:ff:9c:01:15:05:00:6f:09:17:3f:80:02:01
:00:08:09:0a:0b:0c:0d:0e:0f:10:11:12:13:14:15:16:17:18:19:1a:1b:1c:1d:1e:1f:20:21:22:23:24
:25:26:27:28:29:2a:2b SRC=11.11.11.12 DST=11.11.11.2 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=0
DF PROTO=ICMP TYPE=8 CODE=0 ID=277 SEQ=1280
```

The following example is an output ACL log dump:

```
Jul 17 20:38:44 excal-2
%KERN-7-SYSTEM_MSG:
%IPACL-7-DENY:IN= OUT=vsan1 SRC=11.11.11.2 DST=11.11.11.12 LEN=84 TOS=0x00 PREC=0x00
TTL=255 ID=38095 PROTO=ICMP TYPE=0 CODE=0 ID=277 SEQ=1280
```

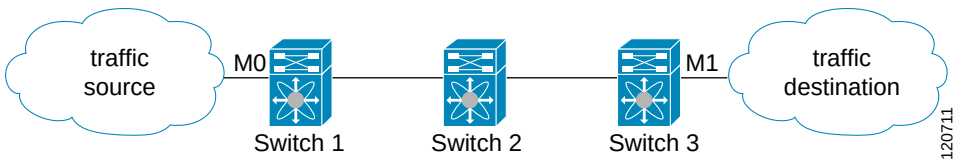
# Applying an IP-ACL to an Interface

You can define IP-ACLs without applying them. However, the IP-ACLs will have no effect until they are applied to an interface on the switch. You can apply IP-ACLs to VSAN interfaces, the management interface, Gigabit Ethernet interfaces on IPS modules and MPS-14/2 modules, and Ethernet PortChannel interfaces.

**Tip** Apply the IP-ACL on the interface closest to the source of the traffic.

When you are trying to block traffic from source to destination, you can apply an inbound IPv4-ACL to M0 on Switch 1 instead of an outbound filter to M1 on Switch 3 (see Figure 5-1).

***Figure 5-1 Denying Traffic on the Inbound Interface***




The **access-group** option controls access to an interface. Each interface can only be associated with one IP-ACL per direction. The ingress direction can have a different IP-ACL than the egress direction. The IP-ACL becomes active when applied to the interface.

**Tip** Create all conditions in an IP-ACL before applying it to the interface.

**Caution** If you apply an IP-ACL to an interface before creating it, all packets in that interface are dropped because the IP-ACL is empty.

The terms *in, out, source*, and *destination* are used as referenced by the switch:

- In—Traffic that arrives at the interface and goes through the switch; the source is where it transmitted from and the destination is where it is transmitted to (on the other side of the router).

**Tip** The IP-ACL applied to the interface for the ingress traffic affects both local and remote traffic.

- Out—Traffic that has already been through the switch and is leaving the interface; the source is where it transmitted from and the destination is where it is transmitted to.

**Tip** The IP-ACL applied to the interface for the egress traffic only affects local traffic.

To apply an IPv4-ACL to an interface, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# interface mgmt0`<br>`switch(config-if)#` | Configures a management interface (mgmt0). |
| **Step 3** | `switch(config-if)# ip access-group restrict_mgmt` | Applies an IPv4-ACL called restrict_mgmt for both the ingress and egress traffic (default). |
| | `switch(config-if)# no ip access-group NotRequired` | Removes the IPv4-ACL called NotRequired. |
| **Step 4** | `switch(config-if)# ip access-group restrict_mgmt in` | Applies an IPv4-ACL called restrict_mgmt (if it does not already exist) for ingress traffic. |
| | `switch(config-if)# no ip access-group restrict_mgmt in` | Removes the IPv4-ACL called restrict_mgmt for ingress traffic. |
| | `switch(config-if)# ip access-group SampleName2 out` | Applies an IPv4-ACL called SampleName2 (if it does not already exist) for egress traffic. |
| | `switch(config-if)# no ip access-group SampleName2 out` | Removes the IPv4-ACL called SampleName2 for egress traffic. |

To apply an IPv6-ACL to an interface, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# interface mgmt0`<br>`switch(config-if)#` | Configures a management interface (mgmt0). |

| | Command | Purpose |
|---|---|---|
| Step 3 | `switch(config-if)# ipv6 traffic-filter RestrictMgmt in` | Applies an IPv6-ACL called RestrictMgmt (if it does not already exist) for ingress traffic. |
| | `switch(config-if)# no ipv6 traffic-filter RestrictMgmt in` | Removes the IPv6-ACL called RestrictMgmt for ingress traffic. |
| | `switch(config-if)# ipv6 traffic-filter SampleName2 out` | Applies an IPv6-ACL called SampleName2 (if it does not already exist) for egress traffic. |
| | `switch(config-if)# no ipv6 traffic-filter SampleName2 out` | Removes the IPv6-ACL called SampleName2 for egress traffic. |

# Applying an IP-ACL to mgmt0

A system default ACL called mgmt0 exists on the mgmt0 interface. This ACL is not visible to the user, so mgmt0 is a reserved ACL name that cannot be used. The mgmt0 ACL blocks most ports and only allows access to required ports in compliance to accepted security policies.

# Verifying Interface IP-ACL Configuration

Use the **show interface** command to display the IPv4-ACL configuration on an interface.

```
switch# show interface mgmt 0
mgmt0 is up
    Internet address(es):
        10.126.95.180/24
        2001:420:54ff:a4::222:5dd/119
        fe80::eaed:f3ff:fee5:d28f/64
    Hardware is GigabitEthernet
    Address is e8ed.f3e5.d28f
    MTU 1500 bytes, BW 1000 Mbps full Duplex
    5144246 packets input, 1008534481 bytes
      2471254 multicast frames, 0 compressed
      0 input errors, 0 frame
      0 overrun, 0 fifo
    1765722 packets output, 1571361034 bytes
      0 underruns, 0 output errors
      0 collisions, 0 fifo
      0 carrier errors
```

Use the **show interface** command to display the IPv6-ACL configuration on an interface.

```
switch# show interface gigabitethernet 2/1
GigabitEthernet2/1 is up
    Hardware is GigabitEthernet, address is 000e.38c6.28b0
    Internet address is 10.1.1.10/24
    MTU 1500  bytes
    Port mode is IPS
    Speed is 1 Gbps
    Beacon is turned off
    Auto-Negotiation is turned on
    ip access-group RestrictMgmt
    5 minutes input rate 1208 bits/sec, 151 bytes/sec, 2 frames/sec
    5 minutes output rate 80 bits/sec, 10 bytes/sec, 0 frames/sec
    6232 packets input, 400990 bytes
      0 multicast frames, 0 compressed
      0 input errors, 0 frame, 0 overrun 0 fifo
```

```
    503 packets output, 27054 bytes, 0 underruns
      0 output errors, 0 collisions, 0 fifo
      0 carrier errors
```

# IP-ACL Counter Cleanup

Use the **clear** command to clear the counters for a specified IPv4-ACL filter entry.

**Note** You cannot use this command to clear the counters for individual filters.

```
switch# show ip access-list abc
ip access-list abc permit tcp any any (0 matches)
ip access-list abc permit udp any any (0 matches)
ip access-list abc permit icmp any any (0 matches)
ip access-list abc permit ip 10.1.1.0 0.0.0.255 (2 matches)
ip access-list abc permit ip 10.3.70.0 0.0.0.255 (7 matches)

switch# clear ip access-list counters abc

switch# show ip access-list abc
ip access-list abc permit tcp any any (0 matches)
ip access-list abc permit udp any any (0 matches)
ip access-list abc permit icmp any any (0 matches)
ip access-list abc permit ip 10.1.1.0 0.0.0.255 (0 matches)
ip access-list abc permit ip 10.3.70.0 0.0.0.255 (0 matches)
```

Use the **clear ipv6 access-list** command to clear the counters for all IPv6-ACLs.

```
switch# clear ipv6 access-list
```

Use the **clear ipv6 access-list** *name* command to clear the counters for a specified IPv6-ACL.

```
switch# clear ipv6 access-list List1
```

**Note** You cannot use this command to clear the counters for each individual filter.


# Configuring Certificate Authorities and Digital Certificates

This chapter includes the following sections:



## About CAs and Digital Certificates

Public Key Infrastructure (PKI) support provides the means for the Cisco MDS 9000 Family switches to obtain and use digital certificates for secure communication in the network. PKI support provides manageability and scalability for IPsec/IKE and SSH.

CAs manage certificate requests and issue certificates to participating entities such as hosts, network devices, or users. The CAs provide centralized key management for the participating entities.

Digital signatures, based on public key cryptography, digitally authenticate devices and individual users. In public key cryptography, such as the RSA encryption system, each device or user has a key-pair containing both a private key and a public key. The private key is kept secret and is known only to the owning device or user only. However, the public key is known to everybody. The keys act as complements. Anything encrypted with one of the keys can be decrypted with the other. A signature is formed when data is encrypted with a sender's private key. The receiver verifies the signature by decrypting the message with the sender's public key. This process relies on the receiver having a copy of the sender's public key and knowing with a high degree of certainty that it really does belong to the sender and not to someone pretending to be the sender.

This section provides information about certificate authorities (CAs) and digital certificates, and includes the following topics:

# Purpose of CAs and Digital Certificates

CAs manage certificate requests and issue certificates to participating entities such as hosts, network devices, or users. The CAs provide centralized key management for the participating entities.

Digital signatures, based on public key cryptography, digitally authenticate devices and individual users. In public key cryptography, such as the RSA encryption system, each device or user has a key-pair containing both a private key and a public key. The private key is kept secret and is known only to the owning device or user only. However, the public key is known to everybody. The keys act as complements. Anything encrypted with one of the keys can be decrypted with the other. A signature is formed when data is encrypted with a sender's private key. The receiver verifies the signature by decrypting the message with the sender's public key. This process relies on the receiver having a copy of the sender's public key and knowing with a high degree of certainty that it really does belong to the sender and not to someone pretending to be the sender.

Digital certificates link the digital signature to the sender. A digital certificate contains information to identify a user or device, such as the name, serial number, company, department, or IP address. It also contains a copy of the entity's public key. The certificate is itself signed by a CA, a third party that is explicitly trusted by the receiver to validate identities and to create digital certificates.

To validate the signature of the CA, the receiver must first know the CA's public key. Normally this process is handled out-of-band or through an operation done at installation. For instance, most web browsers are configured with the public keys of several CAs by default. The Internet Key Exchange (IKE), an essential component of IPsec, can use digital signatures to scalably authenticate peer devices before setting up security associations.

# Trust Model, Trust Points, and Identity CAs

The trust model used in PKI support is hierarchical with multiple configurable trusted CAs. Each participating entity is configured with a list of CAs to be trusted so that the peer's certificate obtained during the security protocol exchanges can be verified, provided it has been issued by one of the locally trusted CAs. To accomplish this, the CA's self-signed root certificate (or certificate chain for a subordinate CA) is locally stored. The process of securely obtaining a trusted CA's root certificate (or the entire chain in the case of a subordinate CA) and storing it locally is called *CA authentication* and is a mandatory step in trusting a CA.

The information about a trusted CA that is locally configured is called the *trust point* and the CA itself is called a *trust point CA*. This information consists of CA certificate (or certificate chain in case of a subordinate CA) and the certificate revocation checking information.

The MDS switch can also enroll with a trust point to obtain an identity certificate (for example, for IPsec/IKE). This trust point is called an *identity CA*.

# RSA Key-Pairs and Identity Certificates

You can generate one or more RSA key-pairs and associate each RSA key-pair with a trust point CA where the MDS switch intends to enroll to obtain an identity certificate. The MDS switch needs only one identity per CA, which consists of one key-pair and one identity certificate per CA.

Cisco MDS NX-OS allows you to generate RSA key-pairs with a configurable key size (or modulus). The default key size is 512. You can also configure an RSA key-pair label. The default key label is the switch fully qualified domain name (FQDN).

The following list summarizes the relationship between trust points, RSA key-pairs, and identity certificates:

- A trust point corresponds to a specific CA that the MDS switch trusts for peer certificate verification for any application (such as IKE or SSH).
- An MDS switch can have many trust points and all applications on the switch can trust a peer certificate issued by any of the trust point CAs.
- A trust point is not restricted to a specific application.
- An MDS switch enrolls with the CA corresponding to the trust point to obtain an identity certificate. You can enroll your switch with multiple trust points thereby obtaining a separate identity certificate from each trust point. The identity certificates are used by applications depending upon the purposes specified in the certificate by the issuing CA. The purpose of a certificate is stored in the certificate as certificate extensions.
- When enrolling with a trust point, you must specify an RSA key-pair to be certified. This key-pair must be generated and associated to the trust point before generating the enrollment request. The association between the trust point, key-pair, and identity certificate is valid until it is explicitly removed by deleting the certificate, key-pair, or trust point.
- The subject name in the identity certificate is the fully qualified domain name for the MDS switch.
- You can generate one or more RSA key-pairs on a switch and each can be associated to one or more trust points. But no more than one key-pair can be associated to a trust point, which means only one identity certificate is allowed from a CA.
- If multiple identity certificates (each from a distinct CA) have been obtained, the certificate that an application selects to use in a security protocol exchange with a peer is application specific.
- You do not need to designate one or more trust points for an application. Any application can use any certificate issued by any trust point as long as the certificate purpose satisfies the application requirements.
- You do not need more than one identity certificate from a trust point or more than one key-pair to be associated to a trust point. A CA certifies a given identity (name) only once and does not issue multiple certificates with the same subject name. If you need more than one identity certificate for a CA, then define another trust point for the same CA, associate another key-pair to it, and have it certified, provided CA allows multiple certificates with the same subject name.

# Multiple Trusted CA Support

An MDS switch can be configured to trust multiple CAs by configuring multiple trust points and associating each with a distinct CA. With multiple trusted CAs, you do not have to enroll a switch with the specific CA that issued a certificate to a peer. Instead, you configure the switch with multiple trusted CAs that the peer trusts. A switch can then use a configured trusted CA to verify certificates offered by a peer that were not issued by the same CA defined in the identity of the switch.

Configuring multiple trusted CAs allows two or more switches enrolled under different domains (different CAs) to verify the identity of each other when using IKE to set up IPsec tunnels.

# PKI Enrollment Support

Enrollment is the process of obtaining an identity certificate for the switch that is used for applications such as IPsec/IKE or SSH. It occurs between the switch requesting the certificate and the certificate authority.

The PKI enrollment process for a switch involves the following steps:

1. Generate an RSA private and public key-pair on the switch.
2. Generate a certificate request in standard format and forward it to the CA.
3. Manual intervention at the CA server by the CA administrator may be required to approve the enrollment request, when it is received by the CA.
4. Receive the issued certificate back from the CA, signed with the CA's private key.
5. Write the certificate into a nonvolatile storage area on the switch (bootflash).

# Manual Enrollment Using Cut-and-Paste Method

Cisco MDS NX-OS supports certificate retrieval and enrollment using a manual cut-and-paste method. Cut-and-paste enrollment literally means you must cut and paste the certificate requests and resulting certificates between the switch and the CA, as follows:

1. Create an enrollment certificate request, which is displayed in base64-encoded text form.
2. Cut and paste the encoded certificate request text in an e-mail message or in a web form and send it to the CA.
3. Receive the issued certificate (in base64-encoded text form) from the CA in an e-mail message or in a web browser download.
4. Cut and paste the issued certificate to the switch using the certificate import facility.

# Multiple RSA Key-Pair and Identity CA Support

Multiple identity CA support enables the switch to enroll with more than one trust point. This results in multiple identity certificates; each from a distinct CA. This allows the switch to participate in IPsec and other applications with many peers using certificates issued by appropriate CAs that are acceptable to those peers.

The multiple RSA key-pair support feature allows the switch to maintain a distinct key pair for each CA with which it is enrolled. Thus, it can match policy requirements for each CA without conflicting with the requirements specified by the other CAs, such as key length. The switch can generate multiple RSA key-pairs and associate each key-pair with a distinct trust point. Thereafter, when enrolling with a trust point, the associated key-pair is used to construct the certificate request.

## Peer Certificate Verification

The PKI support on an MDS switch provides the means to verify peer certificates. The switch verifies certificates presented by peers during security exchanges pertaining to applications, such as IPsec/IKE and SSH. The applications verify the validity of the peer certificates presented to them. The peer certificate verification process involves the following steps:

- Verifies that the peer certificate is issued by one of the locally trusted CAs.
- Verifies that the peer certificate is valid (not expired) with respect to current time.
- Verifies that the peer certificate is not yet revoked by the issuing CA.

For revocation checking use certificate revocation list (CRL) A trust point uses this method to verify that the peer certificate has not been revoked.

## CRL Downloading, Caching, and Checking Support

Certificate revocation lists (CRLs) are maintained by CAs to give information of prematurely revoked certificates, and the CRLs are published in a repository. The download URL is made public and also specified in all issued certificates. A client verifying a peer's certificate should obtain the latest CRL from the issuing CA and use it to determine if the certificate has been revoked. A client can cache the CRLs of some or all of its trusted CAs locally and use them later if necessary until the CRLs expire.

Cisco MDS NX-OS allows the manual configuration of pre-downloaded of CRLs for the trust points, and then caches them in the switch bootflash (cert-store). During the verification of a peer certificate by IPsec or SSH, the issuing CA's CRL is consulted only if the CRL has already been cached locally and the revocation checking is configured to use CRL. Otherwise, CRL checking is not performed and the certificate is considered to be not revoked if no other revocation checking methods are configured. This mode of CRL checking is called CRL optional.

## Import and Export Support for Certificates and Associated Key-Pairs

As part of the CA authentication and enrollment process, the subordinate CA certificate (or certificate chain) and identity certificates can be imported in standard PEM (base64) format.

The complete identity information in a trust point can be exported to a file in the password-protected PKCS#12 standard format. It can be later imported to the same switch (for example, after a system crash) or to a replacement switch. The information in a PKCS#12 file consists of the RSA key-pair, the identity certificate, and the CA certificate (or chain).

# Configuring CAs and Digital Certificates

This section describes the tasks you must perform to allow CAs and digital certificates your Cisco MDS switch device to interoperate. This section includes the following sections:

# Configuring the Host Name and IP Domain Name

You must configure the host name and IP domain name of the switch if they are not already configured. This is required because switch FQDN is used as the subject in the identity certificate. Also, the switch FQDN is used as a default key label when none is specified during key-pair generation. For example, a certificate named SwitchA.example.com is based on a switch host name of SwitchA and a switch IP domain name of example.com.

**Caution** Changing the host name or IP domain name after generating the certificate can invalidate the certificate.

To configure the host name and IP domain name of the switch, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config terminal`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# hostname SwitchA` | Configures the host name (SwitchA) of the switch. |
| **Step 3** | `SwitchA(config)# ip domain-name example.com` | Configures the IP domain name (example.com) of the switch. |

# Generating an RSA Key-Pair

RSA key-pairs are used to sign and/or encrypt and decrypt the security payload during security protocol exchanges for applications such as IKE/IPsec and SSH, and they are required before you can obtain a certificate for your switch.

To generate an RSA key-pair, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config terminal`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# crypto key generate rsa` | Generates an RSA key-pair with the switch FQDN as the default label and 512 as the default modulus. By default, the key is not exportable.<br>**Note** The security policy (or requirement) at the local site (MDS switch) and at the CA (where enrollment is planned) are considered in deciding the appropriate key modulus.<br>**Note** The maximum number of key-pairs you can configure on a switch is 16. |
| | `switch(config)# crypto key generate rsa label SwitchA modulus 768` | Generates an RSA key-pair with the label SwitchA and modulus 768. Valid modulus values are 512, 768, 1024, 1536, and 2048. By default, the key is not exportable. |
| | `switch(config)# crypto key generate rsa exportable` | Generates an RSA key-pair with the switch FQDN as the default label and 512 as the default modulus. The key is exportable.<br>**Caution** The exportability of a key-pair cannot be changed after key-pair generation.<br>**Note** Only exportable key-pairs can be exported in PKCS#12 format. |

# Creating a Trust Point CA Association

To create a trust point CA association, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch(config)# crypto ca trustpoint admin-ca`<br>`switch(config-trustpoint)#` | Declares a trust point CA that the switch should trust and enters trust point configuration submode.<br>**Note** The maximum number of trust points you can declare on a switch is 16. |
| | `switch(config)# no crypto ca trustpoint admin-ca` | Removes the trust point CA. |
| **Step 2** | `switch(config-trustpoint)# enroll terminal` | Specifies manual cut-and-paste certificate enrollment (default).<br>**Note** Manual cut-and-paste certificate enrollment is the only method supported for enrollment. |
| **Step 3** | `switch(config-trustpoint)# rsakeypair SwitchA` | Specifies the label of the RSA key-pair to be associated to this trust point for the purpose of enrollment. It was generated earlier in the "Generating an RSA Key-Pair" section on page 6-136. Only one RSA key-pair can be specified per CA. |
| | `switch(config-trustpoint)# no rsakeypair SwitchA` | Disassociates the RSA key-pair from the trust point (default). |
| **Step 4** | `switch(config-trustpoint)# end`<br>`switch#` | Exits trust point configuration submode. |
| **Step 5** | `switch# copy running-config startup-config` | Copies the running configuration to the startup configuration to ensure the configuration is persistent across reboots. |

# Authenticating the CA

The configuration process of trusting a CA is complete only when the CA is authenticated to the MDS switch. The switch must authenticate the CA. It does this by obtaining the self-signed certificate of the CA in PEM format, which contains the public key of the CA. Because the certificate of the CA is self-signed (the CA signs its own certificate) the public key of the CA should be manually authenticated by contacting the CA administrator to compare the fingerprint of the CA certificate.

**Note** If the CA being authenticated is not a self-signed CA (that is, it is a subordinate CA to another CA, which itself may be a subordinate to yet another CA, and so on, finally ending in a self-signed CA), then the full list of the CA certificates of all the CAs in the certification chain needs to be input during the CA authentication step. This is called the *CA certificate chain* of the CA being authenticated. The maximum number of certificates in a CA certificate chain is 10.

To authenticate the certificate of the CA by cutting and pasting the certificate from an e-mail message or a website, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# crypto ca authenticate admin-ca`<br>`input (cut & paste) CA certificate (chain) in PEM format;`<br>`end the input with a line containing only END OF INPUT :`<br>`-----BEGIN CERTIFICATE-----`<br>`MIIC4jCCAoygAwIBAgIQBWDSiay0GZRPSRIljK0ZejANBgkqhkiG9w0BAQUFADCB`<br>`kDEgMB4GCSqGSIb3DQEJARYRYW1hbmRrZUBjaXNjby5jb20xCzAJBgNVBAYTAklO`<br>`MRIwEAYDVQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UE`<br>`ChMFQ2lzY28xEzARBgNVBAsTCm5ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBD`<br>`QTAeFw0wNTA1MDMyMjQ2MzdaFw0wNzA1MDMyMjU1MTdaMIGQMSAwHgYJKoZIhvcN`<br>`AQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UEBhMCSU4xEjAQBgNVBAgTCUth`<br>`cm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4wDAYDVQQKEwVDaXNjbzETMBEG`<br>`A1UECxMKbmV0c3RvcmFnZTESMBAGA1UEAxMJQXBhcm5hIENBMFwwDQYJKoZIhvcN`<br>`AQEBBQADSwAwSAJBAMW/7b3+DXJPANBsIHHzluNccNM87ypyzwuoSNZXOMpeRXXI`<br>`OzyBAgiXT2ASFuUOwQ1iDM8rO/41jf8RxvYKvysCAwEAAaOBvzCBvDALBgNVHQ8E`<br>`BAMCAcYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUJyjyRoMbrCNMRU2OyRhQ`<br>`GgsWbHEwawYDVR0fBGQwYjAuoCygKoYoaHR0cDovL3NzZS0wOC9DZXJ0RW5yb2xs`<br>`L0FwYXJuYSUyMENBLmNybDAwoC6gLIYqZmlsZTovL1xcc3NlLTA4XENlcnRFbnJv`<br>`bGxcQXBhcm5hJTIwQ0EuY3JsMBAGCSsGAQQBgjcVAQQDAgEAMA0GCSqGSIb3DQEB`<br>`BQUAA0EAHv6UQ+8nE399Tww+KaGr0g0NIJaqNgLh0AFcT0rEyuyt/WYGPzksF9Ea`<br>`NBG7E0oN66zex0EOEfG1Vs6mXp1//w==`<br>`-----END CERTIFICATE-----`<br>` END OF INPUT`<br>`Fingerprint(s): MD5`<br>`Fingerprint=65:84:9A:27:D5:71:03:33:9C:12:23:92:38:6F:78:12`<br><br>`Do you accept this certificate? [yes/no]: y` | Prompts you to cut and paste the certificate of the CA. Use the same name that you used when declaring the CA.<br>**Note** The maximum number of trust points you can authenticate to a specific CA is 10. |

**Note** For subordinate CA authentication, the full chain of CA certificates ending in a self-signed CA is required because the CA chain is needed for certificate verification as well as for PKCS#12 format export.

# Configuring Certificate Revocation Checking Methods

During security exchanges with a client (for example, an IKE peer or SSH user), the MDS switch performs the certificate verification of the peer certificate sent by the client and the verification process may involve certificate revocation status checking.

You can use the CRL method for checking revoked sender certificates. You can configure the switch to check the CRL downloaded from the CA (see the "Configuring a CRL" section on page 6-143). Downloading the CRL and checking locally does not generate traffic in your network. However, certificates can be revoked between downloads and your switch would not be aware of the revocation. Using local CRL checking provides the secure method for checking for revoked certificates.

**Note** You must authenticate the CA before configuring certificate revocation checking.

To configure certificate revocation checking methods, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch(config)# crypto ca trustpoint admin-ca`<br>`switch(config-trustpoint)#` | Declares a trust point CA that the switch should trust and enters trust point configuration submode. |
| **Step 2** | `switch(config-trustpoint)# revocation-check crl` | Specifies CRL (default) as the revocation checking method to be employed during verification of peer certificates issued by the same CA as that of this trust point. |
| | `switch(config-trustpoint)# revocation-check none` | Does not check for revoked certificates. |
| | `switch(config-trustpoint)# no revocation-check` | Reverts to default method. |

# Generating Certificate Requests

You must generate a request to obtain identity certificates from the associated trust point CA for each of your switch's RSA key-pairs. You must then cut and paste the displayed request into an e-mail message or in a website form for the CA.

To generate a request for signed certificates from the CA, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config terminal`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# crypto ca enroll admin-ca`<br>`Create the certificate request ..`<br>`Create a challenge password. You will need to verbally provide this password to the CA Administrator in order to revoke your certificate.`<br>`For security reasons your password will not be saved in the configuration.`<br>`Please make a note of it.`<br>`Password:nbv123`<br>`The subject name in the certificate will be: Vegas-1.cisco.com`<br>`Include the switch serial number in the subject name? [yes/no]: no`<br>`Include an IP address in the subject name [yes/no]: yes`<br>`ip address:172.22.31.162`<br>`The certificate request will be displayed...`<br>`-----BEGIN CERTIFICATE REQUEST-----`<br>`MIIBqzCCARQCAQAwHDEaMBgGA1UEAxMRVmVnYXMtMS5jaXNjby5jb20wgZ8wDQYJ`<br>`KoZIhvcNAQEBBQADgY0AMIGJAoGBAL8Y1UAJ2NC7jUJ1DVaSMqNIgJ2kt8rl4lKY`<br>`0JC6ManNy4qxk8VeMXZSiLJ4JgTzKWdxbLDkTTysnjuCXGvjb+wj0hEhv/y51T9y`<br>`P2NJJ8ornqShrvFZgC7ysN/PyMwKcgzhbVpj+rargZvHtGJ91XTq4WoVkSCzXv8S`<br>`VqyH0vEvAgMBAAGgTzAVBgkqhkiG9w0BCQcxCBMGbmJ2MTIzMDYGCSqGSIb3DQEJ`<br>`DjEpMCcwJQYDVR0RAQH/BBswGYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwDQYJ`<br>`KoZIhvcNAQEEBQADgYEAkT60KER6Qo8nj0sDXZVHSfJZh6K6JtDz3Gkd99GlFWgt`<br>`PftrNcWUE/pw6HayfQl2T3ecgNwel2d15133YBF2bktExiI6Ul88nTOjglXMjja8`<br>`8a23bNDpNsM8rklwA6hWkrVL8NUZEFJxqbjfngPNTZacJCUS6ZqKCMetbKytUx0=`<br>`-----END CERTIFICATE REQUEST-----` | Generates a certificate request for an authenticated CA.<br>**Note** The challenge password is not saved with the configuration. This password is required in the event that your certificate needs to be revoked, so you must remember this password. |

# Installing Identity Certificates

You receive the identity certificate from the CA by e-mail or through a web browser in base64 encoded text form. You must install the identity certificate from the CA by cutting and pasting the encoded textusing the CLI import facility.

To install an identity certificate received from the CA by e-mail or through a web browser, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config terminal`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# crypto ca import admin-ca certificate`<br>`input (cut & paste) certificate in PEM format:`<br>`-----BEGIN CERTIFICATE-----`<br>`MIIEADCCA6qgAwIBAgIKCjOOoQAAAAAAdDANBgkqhkiG9w0BAQUFADCBkDEgMB4G`<br>`CSqGSIb3DQEJARYRYW1hbmRrZUBjaXNjby5jb20xCzAJBgNVBAYTAklOMRIwEAYD`<br>`VQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UEChMFQ2lz`<br>`Y28xEzARBgNVBAsTCm51dHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBDQTAeFw0w`<br>`NTExMTIwMzAyNDBaFw0wNjExMTIwMzEyNDBaMBwxGjAYBgNVBAMTEVZlZ2FzLTEu`<br>`Y2lzY28uY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC/GNVACdjQu41C`<br>`dQ1WkjKjSICdpLfK5eJSmNCQujGpzcuKsZPFXjF2UoiyeCYE8ylncWyw5E08rJ47`<br>`glxr42/sI9IRIb/8udU/cj9jSSfKK56koa7xWYAu8rDfz8jMCnIM4W1aY/q2q4Gb`<br>`x7RifdV06uFqFZEgs17/Elash9LxLwIDAQABo4ICEzCCAg8wJQYDVR0RAQH/BBsw`<br>`GYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwHQYDVR0OBBYEFKCLi+2sspWEfgrR`<br>`bhWmlVyo9jngMIHMBgNVHSMEgcQwgcGAFCco8kaDG6wjTEVNjskYUBoLFmxxoYGW`<br>`pIGTMIGQMSAwHgYJKoZIhvcNAQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UE`<br>`BhMCSU4xEjAQBgNVBAgTCUthcm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4w`<br>`DAYDVQQKEwVDaXNjbzETMBEGA1UECxMKbmV0c3RvcmFnZTESMBAGA1UEAxMJQXBh`<br>`cm5hIENBghAFYNKJrLQZlE9JEiWMrRl6MGsGA1UdHwRkMGIwLqAsoCqGKGh0dHA6`<br>`Ly9zc2UtMDgvQ2VydEVucm9sbC9BcGFybmElMjBDQS5jcmwwMKAuoCyGKmZpbGU6`<br>`Ly9cXHNzZS0wOFxDZXJ0RW5yb2xsXEFwYXJuYSUyMENBLmNybDCBigYIKwYBBQUH`<br>`AQEEfjB8MDsGCCsGAQUFBzAChi9odHRwOi8vc3NlLTA4L0NlcnRFbnJvbGwvc3Nl`<br>`LTA4X0FwYXJuYSUyMENBLmNydDA9BggrBgEFBQcwAoYxZmlsZTovL1xcc3NlLTA4`<br>`XENlcnRFbnJvbGxcc3NlLTA4X0FwYXJuYSUyMENBLmNydDANBgkqhkiG9w0BAQUF`<br>`AANBADbGBGsbe7GNLh9xeOTWBNbm24U69ZSuDDcOcUZUUTgrpnTqVpPyejtsyflw`<br>`E36cIZu4WsExREqxbTk8ycx7V5o=`<br>`-----END CERTIFICATE-----` | Prompts you to cut and paste the identity certificate for the CA named admin-ca.<br>**Note** The maximum number of identify certificates you can configure on a switch is 16. |

# Saving Your Configuration

Save your work when you make configuration changes or the information is lost when you exit.

# Ensuring Trust Point Configurations Persist Across Reboots

The trust point configuration is a normal Cisco NX-OS configuration that persists across system reboots only if you copy it explicitly to the startup configuration. The certificates, key-pairs, and CRL associated with a trust point are automatically persistent if you have already copied the trust point configuration in the startup configuration. Conversely, if the trust point configuration is not copied to the startup configuration, the certificates, key-pairs, and CRL associated with it are not persistent since they require the corresponding trust point configuration after a reboot. Always copy the running configuration to the startup configuration to ensure that the configured certificates, key-pairs, and CRLs are persistent. Also, save the running configuration after deleting a certificate or key-pair to ensure that the deletions are permanent.

The certificates and CRL associated with a trust point automatically become persistent when imported (that is, without an explicitly copying to the startup configuration) if the specific trust point is already saved in startup configuration.

We also recommend that you create a password-protected backup of the identity certificates and save it to an external server (see the "Exporting and Importing Identity Information in PKCS#12 Format" section on page 6-142).

**Note** Copying the configuration to an external server does include the certificates and key-pairs.

# Monitoring and Maintaining CA and Certificates Configuration

The tasks in the section are optional. This section includes the following topics:



## Exporting and Importing Identity Information in PKCS#12 Format

You can export the identity certificate along with the RSA key-pair and CA certificate (or the entire chain in the case of a subordinate CA) of a trust point to a PKCS#12 file for backup purposes. You can later import the certificate and RSA key-pair to recover from a system crash on your switch or when you replace the supervisor modules.

**Note** Only the **bootflash:***filename* format local syntax is supported when specifying the export and import URL.

To export a certificate and key-pair to a PKCS#12-formatted file, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config terminal`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# crypto ca export admin-ca pkcs12 bootflash:adminid.p12 nbv123` | Exports the identity certificate and associated key-pair and CA certificates for trust point admin-ca to the file bootflash:adminid.p12 in PKCS#12 format, protected using password nbv123. |
| **Step 3** | `switch(config)# exit`<br>`switch#` | Returns to EXEC mode. |
| **Step 4** | `switch# copy bootflash:adminid.p12 tftp:adminid.p12` | Copies the PKCS#12 format file to a TFTP server. |

To import a certificate and key-pair from a PKCS#12-formatted file, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# copy tftp:adminid.p12 bootflash:adminid.p12` | Copies the PKCS#12 format file from a TFTP server. |
| **Step 2** | `switch# config terminal`<br>`switch(config)#` | Enters configuration mode. |
| **Step 3** | `switch(config)# crypto ca import admin-ca pkcs12 bootflash:adminid.p12 nbv123` | Imports the identity certificate and associated key-pair and CA certificates for trust point admin-ca from the file bootflash:adminid.p12 in PKCS#12 format, protected using password nbv123. |

**Note** The trust point must be empty (with no RSA key-pair associated with it and no CA is associated with it using CA authentication) for the PKCS#12 file import to succeed.

## Configuring a CRL

To import the CRL from a file to a trust point, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# copy tftp:adminca.crl bootflash:adminca.crl` | Downloads the CRL. |
| **Step 2** | `switch# config terminal`<br>`switch(config)#` | Enters configuration mode. |
| **Step 3** | `switch(config)# crypto ca crl request admin-ca bootflash:adminca.crl` | Configures or replaces the current CRL with the one specified in the file. |

## Deleting Certificates from the CA Configuration

You can delete the identity certificates and CA certificates that are configured in a trust point. You must first delete the identity certificate, followed by the CA certificates. After deleting the identity certificate, you can disassociate the RSA key-pair from a trust point. The certificate deletion is necessary to remove expired or revoked certificates, certificates whose key-pairs are compromised (or suspected to be compromised) or CAs that are no longer trusted.

To delete the CA certificate (or the entire chain in the case of a subordinate CA) from a trust point, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# crypto ca trustpoint myCA` | Enters trustpoint configuration submode. |
| **Step 3** | `switch(config-trustpoint)# delete ca-certificate` | Deletes the CA certificate or certificate chain. |

| | Command | Purpose |
|---|---|---|
| **Step 4** | `switch(config-trustpoint)# delete certificate` | Deletes the identity certificate. |
| | `switch(config-trustpoint)# delete certificate force` | Forces the deletion of the identity certificate.<br>**Note** If the identity certificate being deleted is the last-most or only identity certificate in the device, you must use the **force** option to delete it. This ensures that the administrator does not mistakenly delete the last-most or only identity certificate and leave the applications (such as IKE and SSH) without a certificate to use. |
| **Step 5** | `switch(config-trustpoint)# end`<br>`switch#` | Returns to EXEC mode. |
| **Step 6** | `switch# copy running-config startup-config` | Copies the running configuration to the startup configuration to ensure the configuration is persistent across reboots. |

## Deleting RSA Key-Pairs from Your Switch

Under certain circumstances you may want to delete your switch's RSA key-pairs. For example, if you believe the RSA key-pairs were compromised in some way and should no longer be used, you should delete the key-pairs.

To delete RSA key-pairs from your switch, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config terminal`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# crypto key zeroize rsa MyKey` | Deletes the RSA key-pair whose label is MyKey. |
| **Step 3** | `switch(config)# end`<br>`switch#` | Returns to EXEC mode. |
| **Step 4** | `switch# copy running-config startup-config` | Copies the running configuration to the startup configuration to ensure the configuration is persistent across reboots. |

**Note** After you delete RSA key-pairs from a switch, ask the CA administrator to revoke your switch's certificates at the CA. You must supply the challenge password you created when you originally requested the certificates. See "Generating Certificate Requests" section on page 6-140.

### Displaying Key-Pair and CA Information

To view key-pair and CA information, use the following commands in EXEC mode:

| Command | Purpose |
|---|---|
| `switch# show crypto key mypubkey rsa` | Displays information about the switch's RSA public keys. |
| `switch# show crypto ca certificates` | Displays information on CA and identity certificates. |
| `switch# show crypto ca crl` | Displays information about CA CRLs. |
| `switch# show crypto ca trustpoints` | Displays information about CA trust points. |

# Example Configurations

This section shows an example of the tasks you can use to configure certificates and CRLs on the Cisco MDS 9000 Family switches using the Microsoft Windows Certificate server.

This section includes the following topics:



## Configuring Certificates on the MDS Switch

To configure certificates on an MDS switch, follow these steps:

**Step 1** Configure the switch FQDN.

```
switch# config t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# switchname Vegas-1
Vegas-1(config)#
```

**Step 2** Configure the DNS domain name for the switch.

```
Vegas-1(config)# ip domain-name cisco.com
Vegas-1(config)#
```

**Step 3** Create a trust point.

```
Vegas-1(config)# crypto ca trustpoint myCA
Vegas-1(config-trustpoint)# exit
Vegas-1(config)# do show crypto ca trustpoints
trustpoint: myCA; key:
revokation methods:  crl
Vegas-1(config)#
```

**Step 4** Create an RSA key-pair for the switch.

```
Vegas-1(config)# crypto key generate rsa label myKey exportable modulus 1024
Vegas-1(config)# do show crypto key mypubkey rsa
key label: myKey
key size: 1024
exportable: yes

Vegas-1(config)#
```

**Step 5** Associate the RSA key-pair to the trust point.

```
Vegas-1(config)# crypto ca trustpoint myCA
Vegas-1(config-trustpoint)# rsakeypair myKey
Vegas-1(config-trustpoint)# exit
Vegas-1(config)# do show crypto ca trustpoints
trustpoint: myCA; key: myKey
revokation methods:  crl
Vegas-1(config)#
```

**Step 6** Download the CA certificate from the Microsoft Certificate Service web interface (see the "Downloading a CA Certificate" section on page 6-148)

**Step 7** Authenticate the CA that you want to enroll to the trust point.

```
Vegas-1(config)# crypto ca authenticate myCA
input (cut & paste) CA certificate (chain) in PEM format;
end the input with a line containing only END OF INPUT :
-----BEGIN CERTIFICATE-----
MIIC4jCCAoygAwIBAgIQBWDSiay0GZRPSRIljK0ZejANBgkqhkiG9w0BAQUFADCB
kDEgMB4GCSqGSIb3DQEJARYRYW1hbmRrZUBjaXNjby5jb20xCzAJBgNVBAYTAklO
MRIwEAYDVQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UE
ChMFQ2lzY28xEzARBgNVBAsTCm5ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBD
QTAeFw0wNTA1MDMyMjQ2MzdaFw0wNzA1MDMyMjU1MTdaMIGQMSAwHgYJKoZIhvcN
AQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UEBhMCSU4xEjAQBgNVBAgTCUth
cm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4wDAYDVQQKEwVDaXNjbzETMBEG
A1UECxMKbmV0c3RvcmFnZTESMBAGA1UEAxMJQXBhcm5hIENBMFwwDQYJKoZIhvcN
AQEBBQADSwAwSAJBAMW/7b3+DXJPANBsIHHzluNccNM87ypyzwuoSNZXOMpeRXXI
OzyBAgiXT2ASFuUOwQ1iDM8rO/41jf8RxvYKvysCAwEAAaOBvzCBvDALBgNVHQ8E
BAMCAcYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUJyjyRoMbrCNMRU2OyRhQ
GgsWbHEwawYDVR0fBGQwYjAuoCygKoYoaHR0cDovL3NzZS0wOC9DZXJ0RW5yb2xs
L0FwYXJuYSUyMENBLmNybDAwoC6gLIYqZmlsZTovL1xcc3NlLTA4XENlcnRFbnJv
bGxcQXBhcm5hJTIwQ0EuY3JsMBAGCSsGAQQBgjcVAQQDAgEAMA0GCSqGSIb3DQEB
BQUAA0EAHv6UQ+8nE399Tww+KaGr0g0NIJaqNgLh0AFcT0rEyuyt/WYGPzksF9Ea
NBG7E0oN66zex0EOEfG1Vs6mXp1//w==
-----END CERTIFICATE-----
 END OF INPUT
Fingerprint(s): MD5 Fingerprint=65:84:9A:27:D5:71:03:33:9C:12:23:92:38:6F:78:12



Do you accept this certificate? [yes/no]:y
Vegas-1(config)#

Vegas-1(config)# do show crypto ca certificates
Trustpoint: myCA
CA certificate 0:
subject= /emailAddress=admin@yourcompany.com/C=IN/ST=Karnataka/L=Bangalore/O=Yourcompany/O
U=netstorage/CN=Aparna CA
issuer= /emailAddress=admin@yourcompany.com/C=IN/ST=Karnataka/L=Bangalore/O=Yourcompany/OU
=netstorage/CN=Aparna CA
serial=0560D289ACB419944F4912258CAD197A
notBefore=May  3 22:46:37 2005 GMT
notAfter=May  3 22:55:17 2007 GMT
MD5 Fingerprint=65:84:9A:27:D5:71:03:33:9C:12:23:92:38:6F:78:12
```

```
purposes: sslserver sslclient ike
```

**Step 8** Generate a request certificate to use to enroll with a trust point.

```
Vegas-1(config)# crypto ca enroll myCA
 Create the certificate request ..
 Create a challenge password. You will need to verbally provide this
  password to the CA Administrator in order to revoke your certificate.
  For security reasons your password will not be saved in the configuration.
  Please make a note of it.
  Password:nbv123
 The subject name in the certificate will be: Vegas-1.cisco.com
 Include the switch serial number in the subject name? [yes/no]:no
 Include an IP address in the subject name [yes/no]:yes
ip address:10.10.1.1
 The certificate request will be displayed...
-----BEGIN CERTIFICATE REQUEST-----
MIIBqzCCARQCAQAwHDEaMBgGA1UEAxMRVmVnYXMtMS5jaXNjby5jb20wgZ8wDQYJ
KoZIhvcNAQEBBQADgY0AMIGJAoGBAL8Y1UAJ2NC7jUJ1DVaSMqNIgJ2kt8rl4lKY
0JC6ManNy4qxk8VeMXZSiLJ4JgTzKWdxbLDkTTysnjuCXGvjb+wj0hEhv/y51T9y
P2NJJ8ornqShrvFZgC7ysN/PyMwKcgzhbVpj+rargZvHtGJ91XTq4WoVkSCzXv8S
VqyH0vEvAgMBAAGgTzAVBgkqhkiG9w0BCQcxCBMGbmJ2MTIzMDYGCSqGSIb3DQEJ
DjEpMCcwJQYDVR0RAQH/BBswGYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwDQYJ
KoZIhvcNAQEEBQADgYEAkT60KER6Qo8nj0sDXZVHSfJZh6K6JtDz3Gkd99GlFWgt
PftrNcWUE/pw6HayfQl2T3ecgNwel2d15133YBF2bktExiI6Ul88nTOjglXMjja8
8a23bNDpNsM8rklwA6hWkrVL8NUZEFJxqbjfngPNTZacJCUS6ZqKCMetbKytUx0=
-----END CERTIFICATE REQUEST-----

Vegas-1(config)#
```

**Step 9** Request an identity certificate from the Microsoft Certificate Service web interface (see the "Requesting an Identity Certificate" section on page 6-152).

**Step 10** Import the identity certificate.

```
Vegas-1(config)# crypto ca import myCA certificate
input (cut & paste) certificate in PEM format:
-----BEGIN CERTIFICATE-----
MIIEADCCA6qgAwIBAgIKCjOOoQAAAAAAdDANBgkqhkiG9w0BAQUFADCBkDEgMB4G
CSqGSIb3DQEJARYRYW1hbmRrZUBjaXNjby5jb20xCzAJBgNVBAYTAklOMRIwEAYD
VQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UEChMFQ2lz
Y28xEzARBgNVBAsTCm5ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBDQTAeFw0w
NTExMTIwMzAyNDBaFw0wNjExMTIwMzEyNDBaMBwxGjAYBgNVBAMTEVZlZ2FzLTEu
Y2lzY28uY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC/GNVACdjQu41C
dQ1WkjKjSICdpLfK5eJSmNCQujGpzcuKsZPFXjF2UoiyeCYE8ylncWyw5E08rJ47
glxr42/sI9IRIb/8udU/cj9jSSfKK56koa7xWYAu8rDfz8jMCnIM4W1aY/q2q4Gb
x7RifdV06uFqFZEgs17/Elash9LxLwIDAQABo4ICEzCCAg8wJQYDVR0RAQH/BBsw
GYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwHQYDVR0OBBYEFKCLi+2sspWEfgrR
bhWmlVyo9jngMIHMBgNVHSMEgcQwgcGAFCco8kaDG6wjTEVNjskYUBoLFmxxoYGW
pIGTMIGQMSAwHgYJKoZIhvcNAQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UE
BhMCSU4xEjAQBgNVBAgTCUthcm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4w
DAYDVQQKEwVDaXNjbzETMBEGA1UECxMKbmV0c3RvcmFnZTESMBAGA1UEAxMJQXBh
cm5hIENBghAFYNKJrLQZlE9JEiWMrRl6MGsGA1UdHwRkMGIwLqAsoCqGKGh0dHA6
Ly9zc2UtMDgvQ2VydEVucm9sbC9BcGFybmElMjBDQS5jcmwwMKAuoCyGKmZpbGU6
Ly9cXHNzZS0wOFxDZXJ0RW5yb2xsXEFwYXJuYSUyMENBLmNybDCBigYIKwYBBQUH
AQEEfjB8MDsGCCsGAQUFBzAChi9odHRwOi8vc3NlLTA4L0NlcnRFbnJvbGwvc3Nl
LTA4X0FwYXJuYSUyMENBLmNydDA9BggrBgEFBQcwAoYxZmlsZTovL1xcc3NlLTA4
XENlcnRFbnJvbGxcc3NlLTA4X0FwYXJuYSUyMENBLmNydDANBgkqhkiG9w0BAQUF
AANBADbGBGsbe7GNLh9xeOTWBNbm24U69ZSuDDcOcUZUUTgrpnTqVpPyejtsyflw
E36cIZu4WsExREqxbTk8ycx7V5o=
-----END CERTIFICATE-----
Vegas-1(config)# exit
Vegas-1#
```

**Step 11** Verify the certificate configuration.

```
Vegas-1# show crypto ca certificates
Trustpoint: myCA
certificate:
subject= /CN=Vegas-1.cisco.com
issuer= /emailAddress=admin@yourcompany.com/C=IN/ST=Karnataka/L=Bangalore/O=Cisco/OU
=netstorage/CN=Aparna CA
serial=0A338EA1000000000074
notBefore=Nov 12 03:02:40 2005 GMT
notAfter=Nov 12 03:12:40 2006 GMT
MD5 Fingerprint=3D:33:62:3D:B4:D0:87:A0:70:DE:A3:87:B3:4E:24:BF
purposes: sslserver sslclient ike

CA certificate 0:
subject= /emailAddress=admin@yourcompany.com/C=IN/ST=Karnataka/L=Bangalore/O=Yourcompany/O
U=netstorage/CN=Aparna CA
issuer= /emailAddress=admin@yourcompany.com/C=IN/ST=Karnataka/L=Bangalore/O=Yourcompany/OU
=netstorage/CN=Aparna CA
serial=0560D289ACB419944F4912258CAD197A
notBefore=May  3 22:46:37 2005 GMT
notAfter=May  3 22:55:17 2007 GMT
MD5 Fingerprint=65:84:9A:27:D5:71:03:33:9C:12:23:92:38:6F:78:12
purposes: sslserver sslclient ike
```

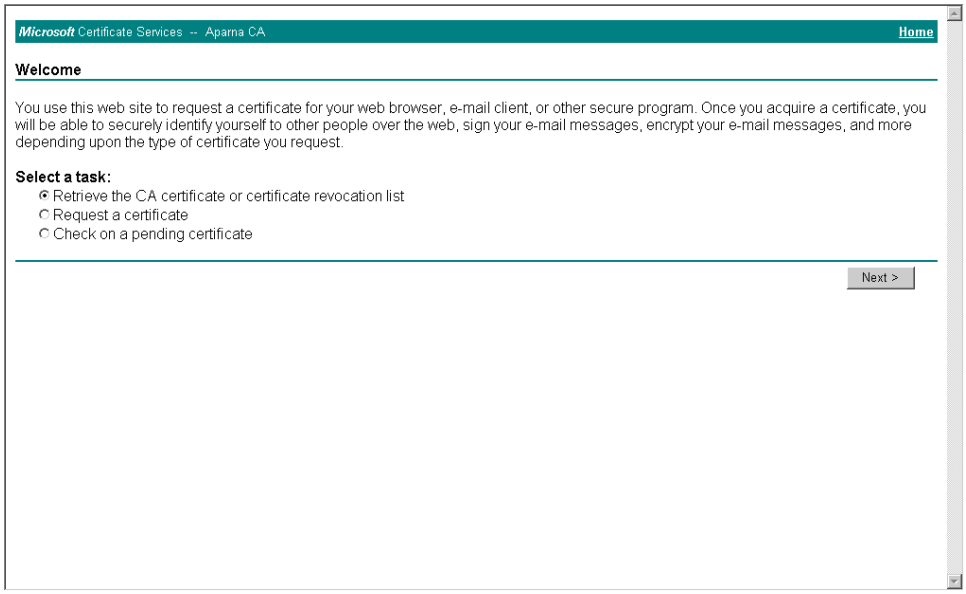
**Step 12** Save the certificate configuration to the startup configuration.

```
Vegas-1# copy running-config startup-config
```

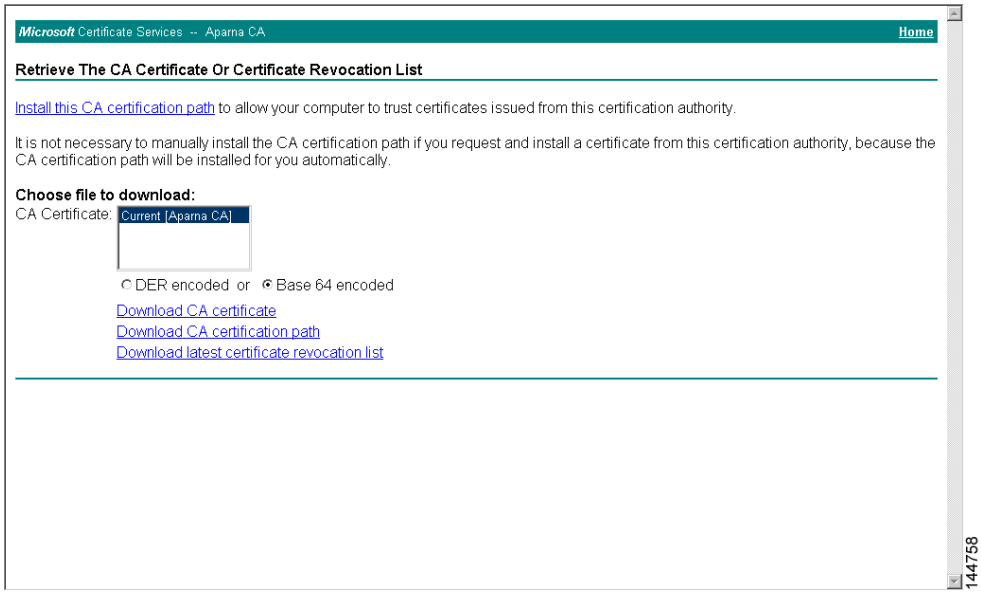
# Downloading a CA Certificate

To download a CA certificate from the Microsoft Certificate Services web interface, follow these steps:

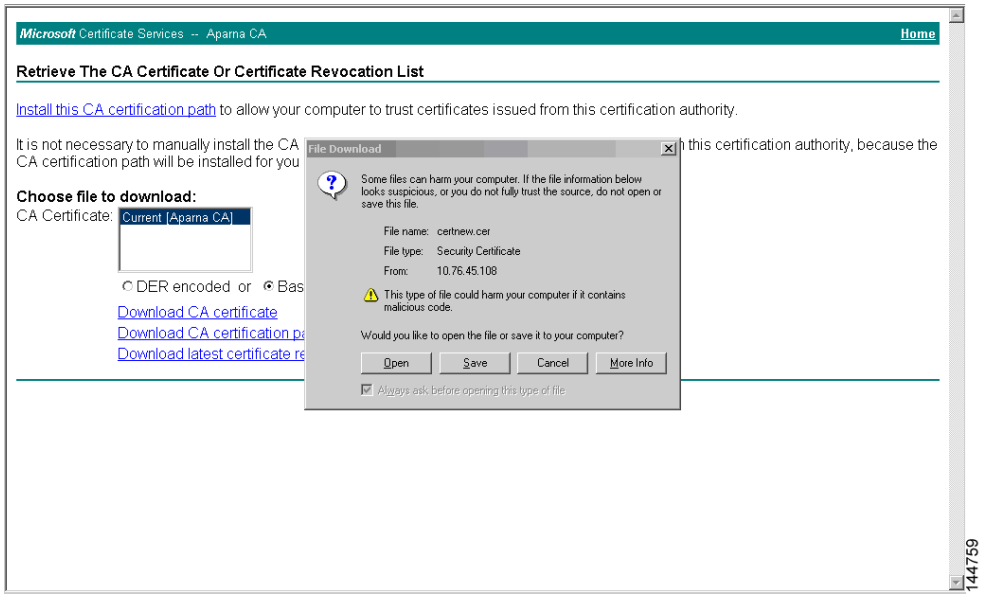
**Step 1** Click the **Retrieve the CA certificate or certificate revocation task** radio button in the Microsoft Certificate Services web interface and click the **Next button**.



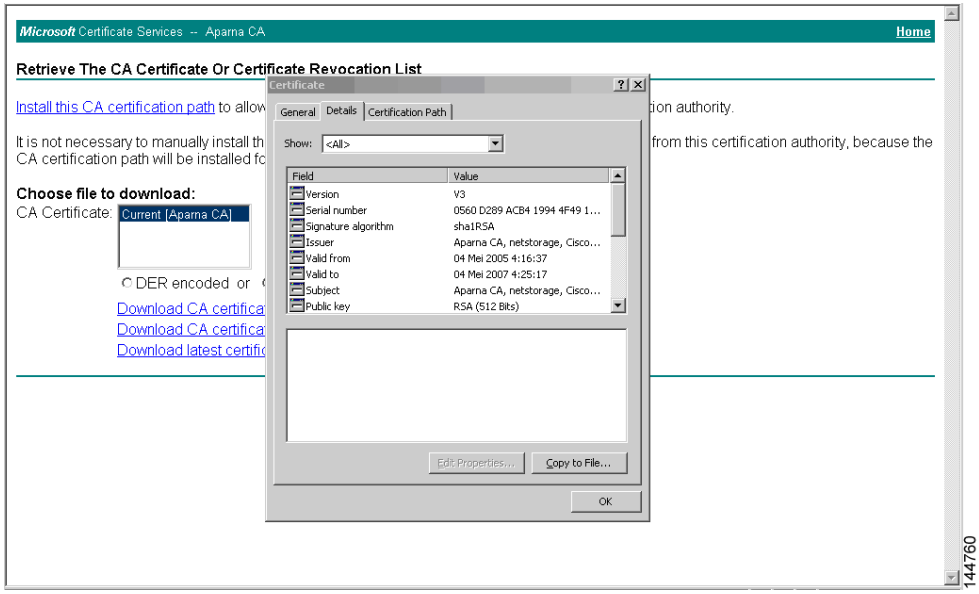

**Step 2** Select the CA certificate file to download from the displayed list. Click the **Base 64 encoded** radio button, and choose the **Download CA certificate** link.



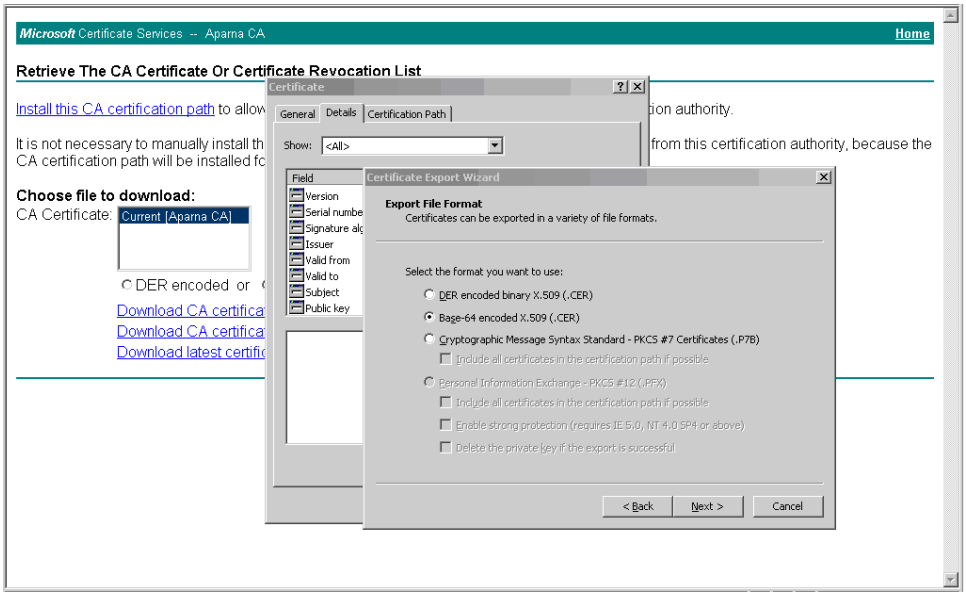

**Step 3** Click the **Open** button in the File Download dialog box.



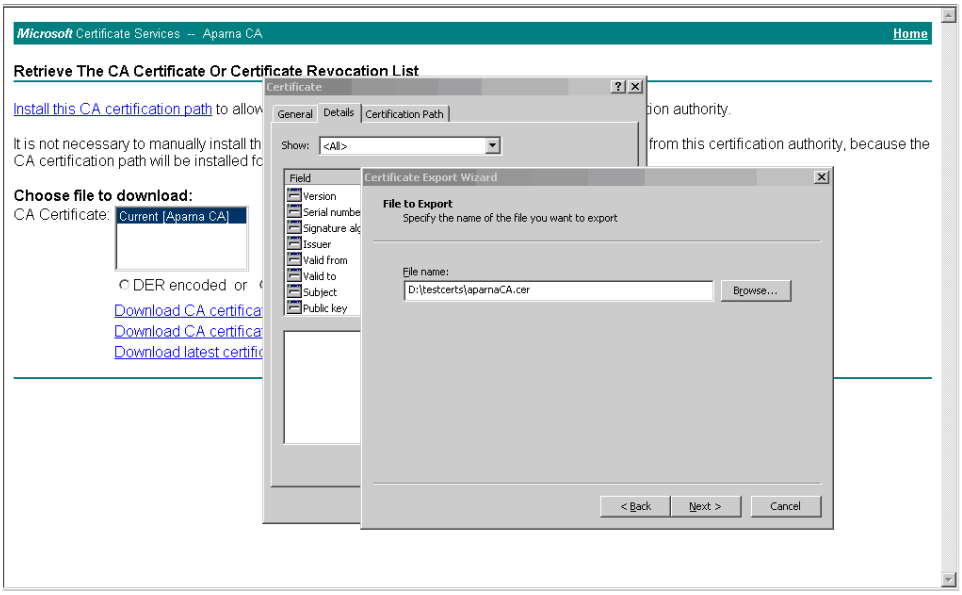

**Step 4** Click the **Copy to File** button in the Certificate dialog box and click **OK**.



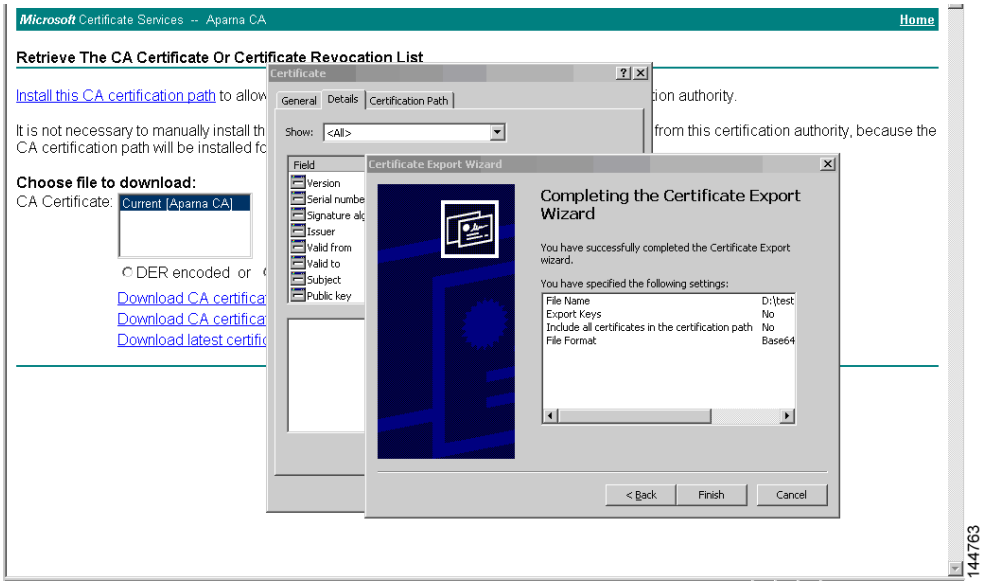

**Step 5** Select the **Base-64 encoded X.509 (CER)** on the Certificate Export Wizard dialog box and click **Next**.



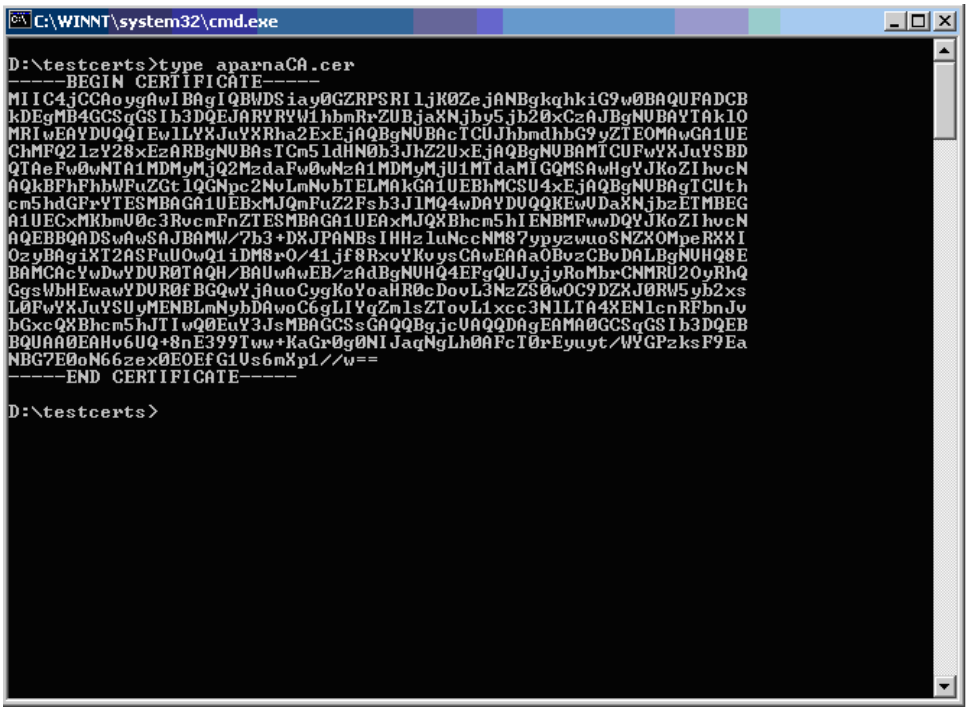

**Step 6** Enter the destination file name in the File name: text box on the Certificate Export Wizard dialog box and click **Next**.




**Step 7** Click the **Finish** button on the Certificate Export Wizard dialog box.



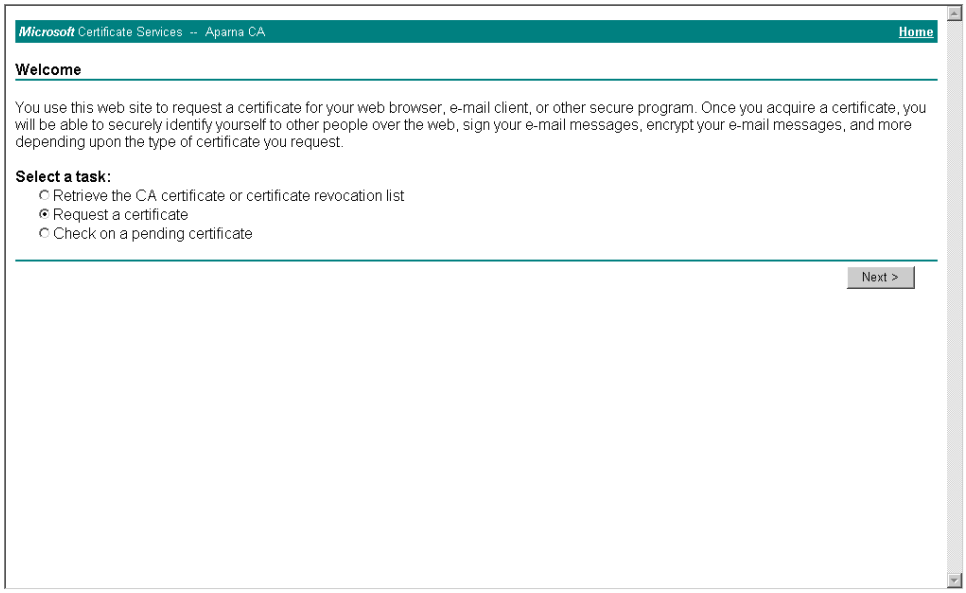

**Step 8** Display the CA certificate stored in Base-64 (PEM) format using the Microsoft Windows **type** command.



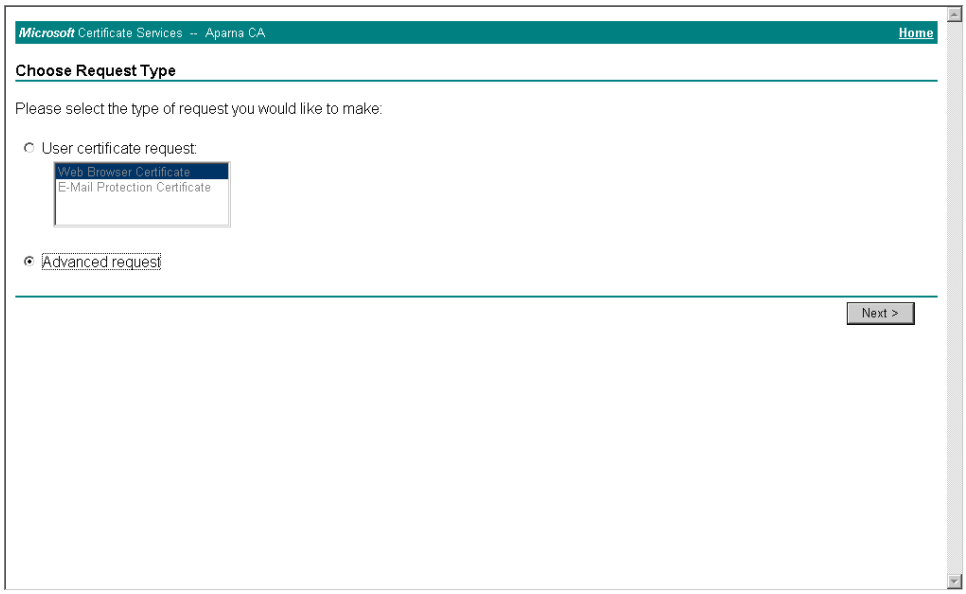

# Requesting an Identity Certificate

To request an identify certificate from a Microsoft Certificate server using a PKCS#10 certificate signing request (CRS), follow these steps:

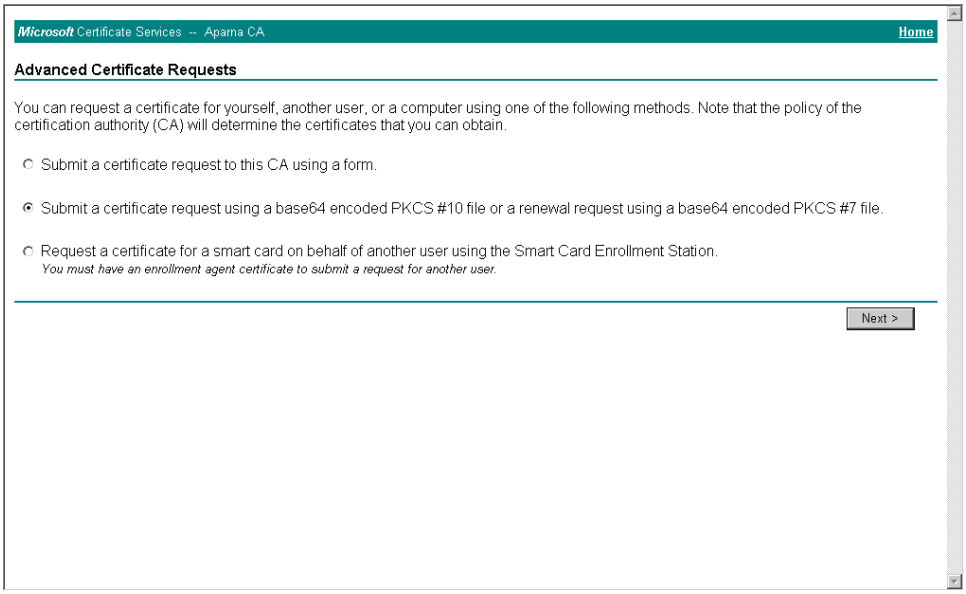
**Step 1** Click the **Request a certificate** radio button on the Microsoft Certificate Services web interface and click **Next**.




**Step 2** Click the **Advanced request** radio button and click **Next**.

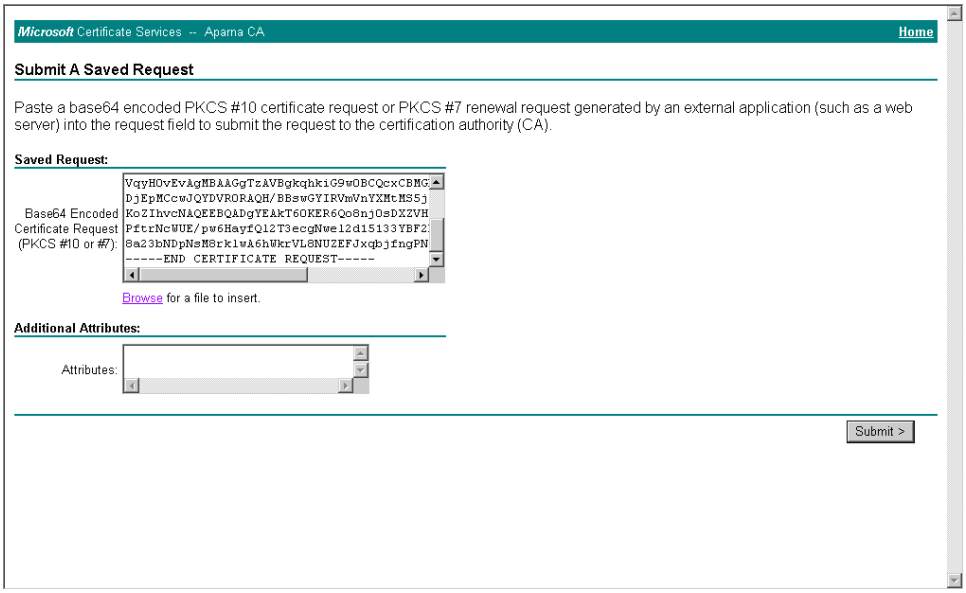



**Step 3** Click the **Submit a certificate request using a base64 encoded PKCS#10 file or a renewal request using a base64 encoded PKCS#7 file** radio button and click **Next**.

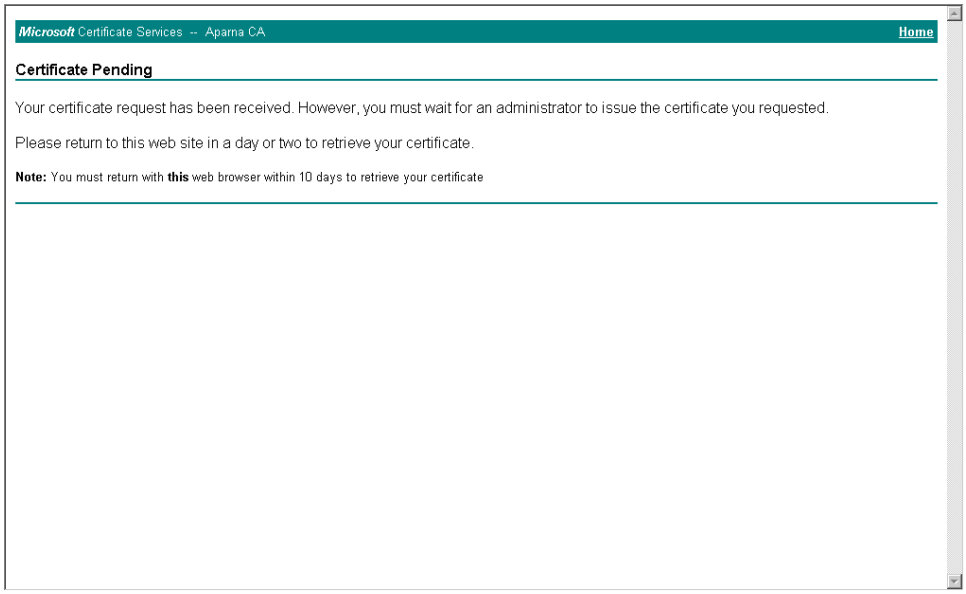



**Step 4** Paste the base64 PKCS#10 certificate request in the Saved Request text box and click **Next**.

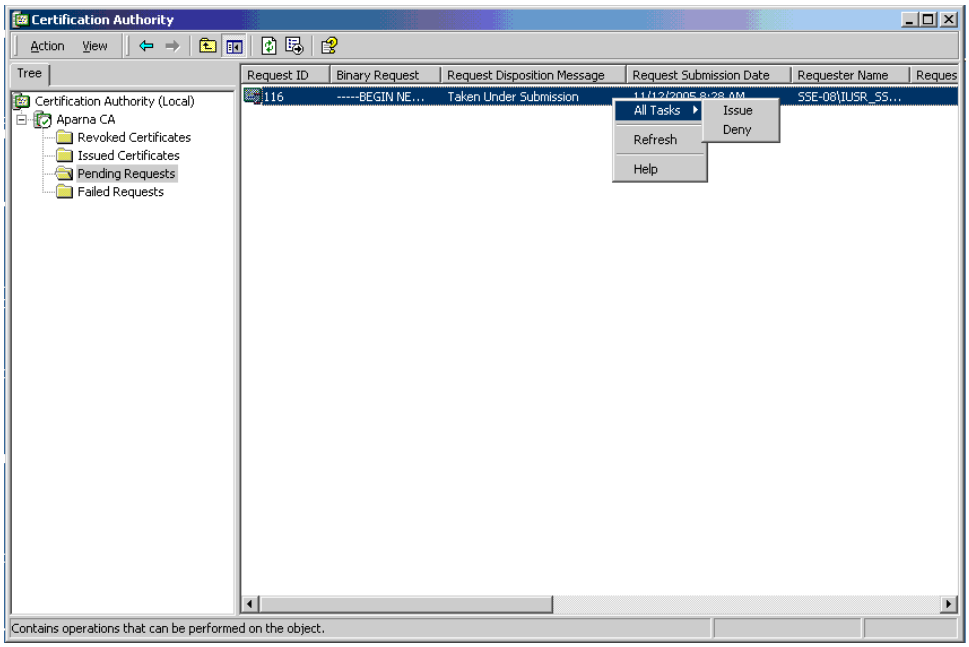
The certificate request is copied from the MDS switch console (see the "Generating Certificate Requests" section on page 6-140 and "Configuring Certificates on the MDS Switch" section on page 6-145).



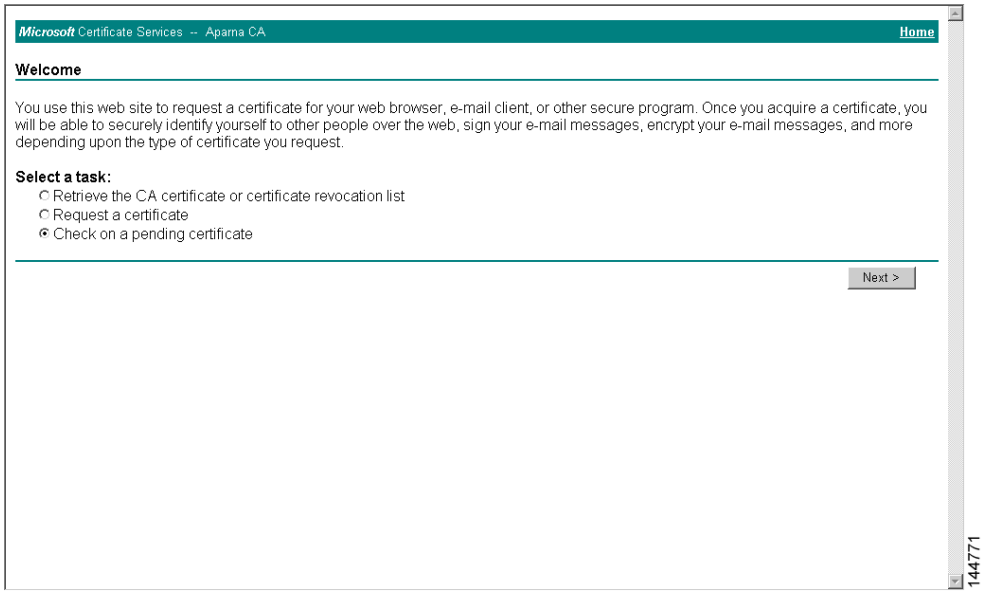

**Step 5** Wait one or two days until the certificate is issued by the CA administrator.



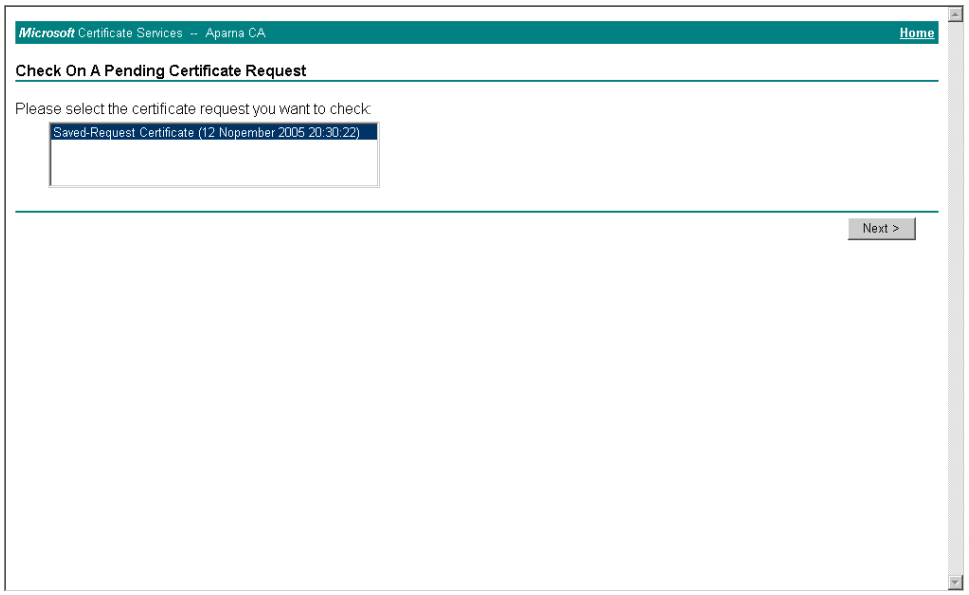

**Step 6** The CA administrator approves the certificate request.



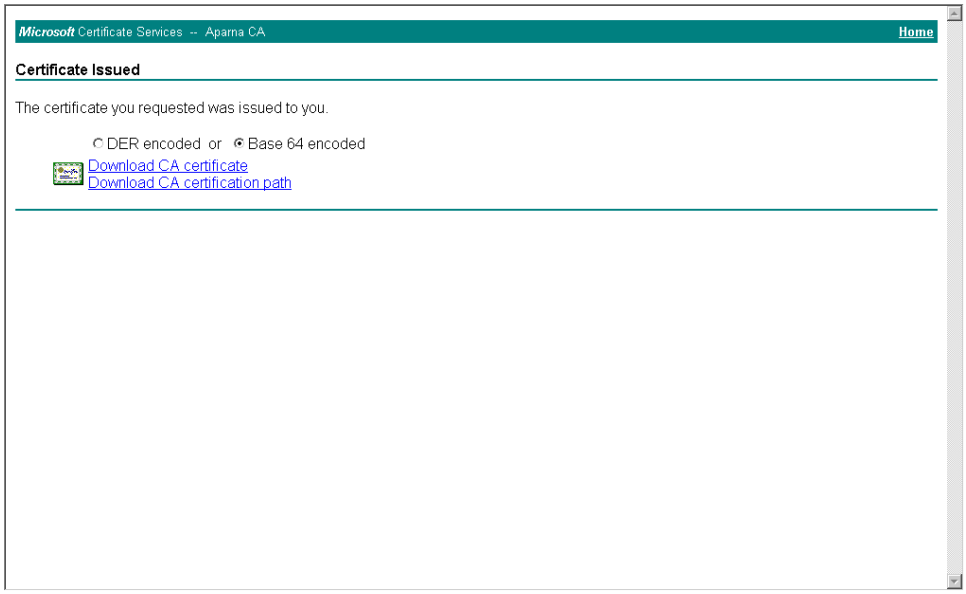

**Step 7** Click the **Check on a pending certificate** radio button on the Microsoft Certificate Services web interface and click **Next**.



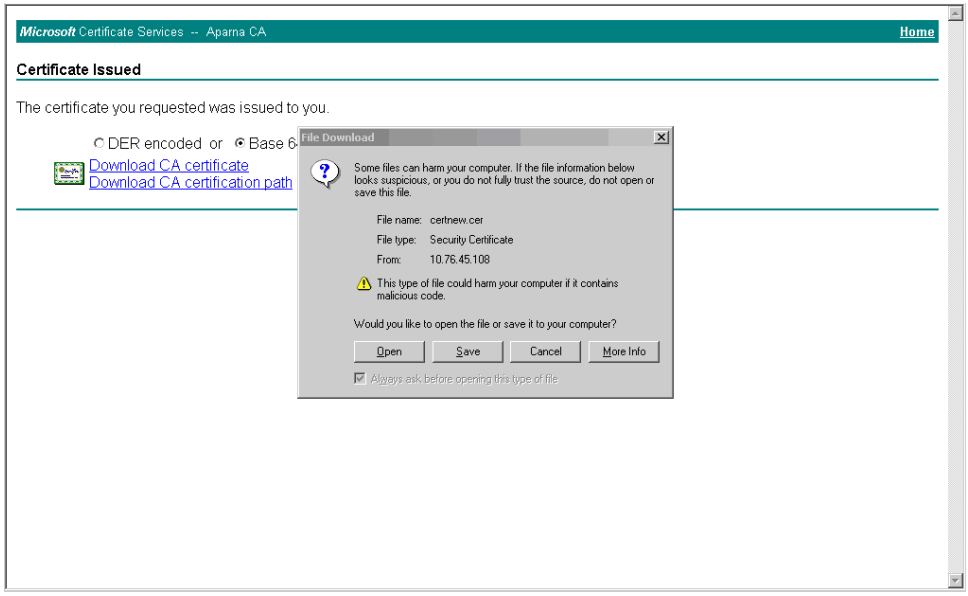

**Step 8** Select the certificate request you want to check and click **Next**.



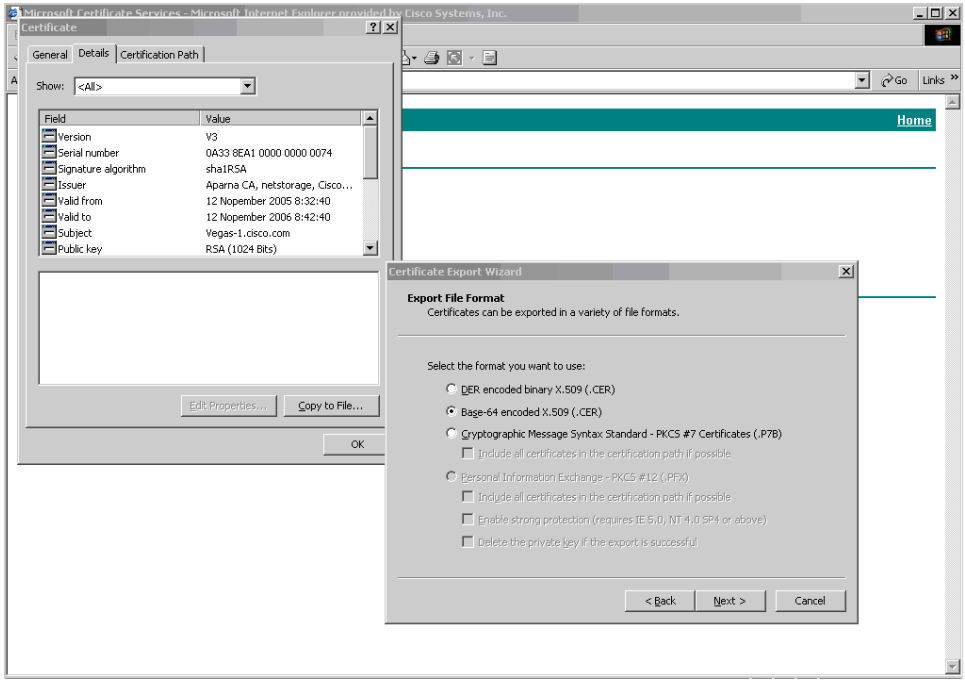

**Step 9** Select **Base 64 encoded** and click the **Download CA certificate** link.



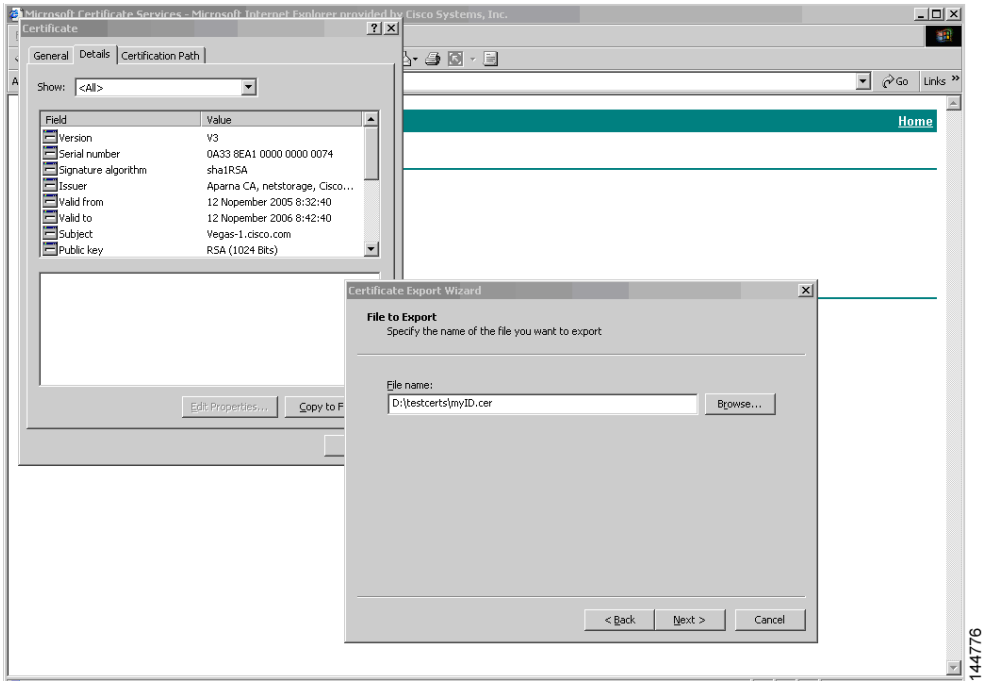

**Step 10** Click **Open** on the File Download dialog box.



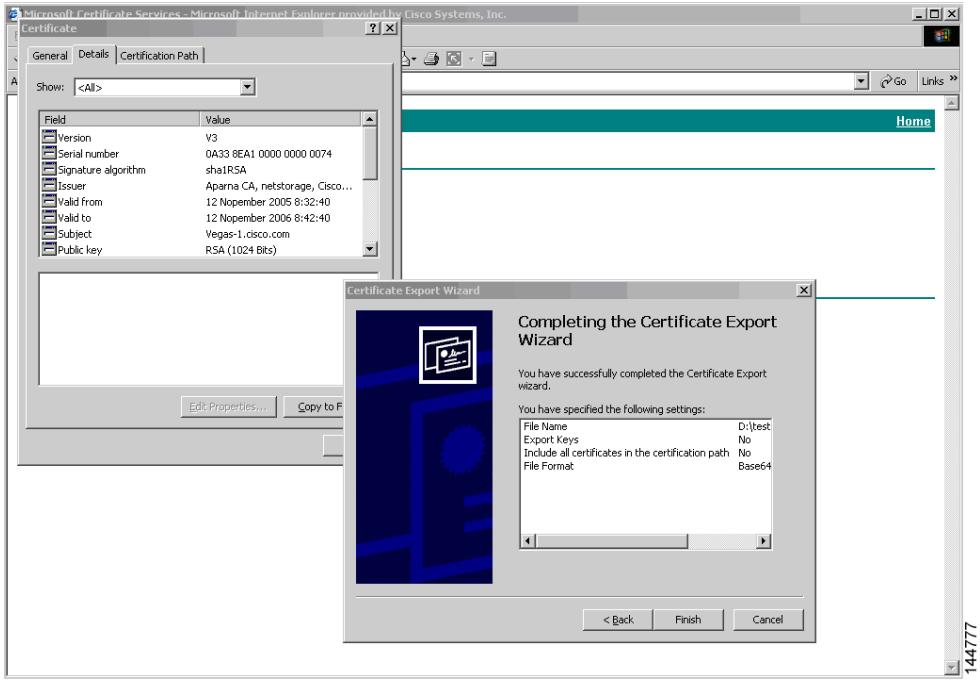

**Step 11** Click the **Details** tab on the Certificate dialog and click the **Copy to File** button. Click the **Base-64 encoded X.509 (.CER)** radio button on the Certificate Export Wizard dialog box and click **Next**.



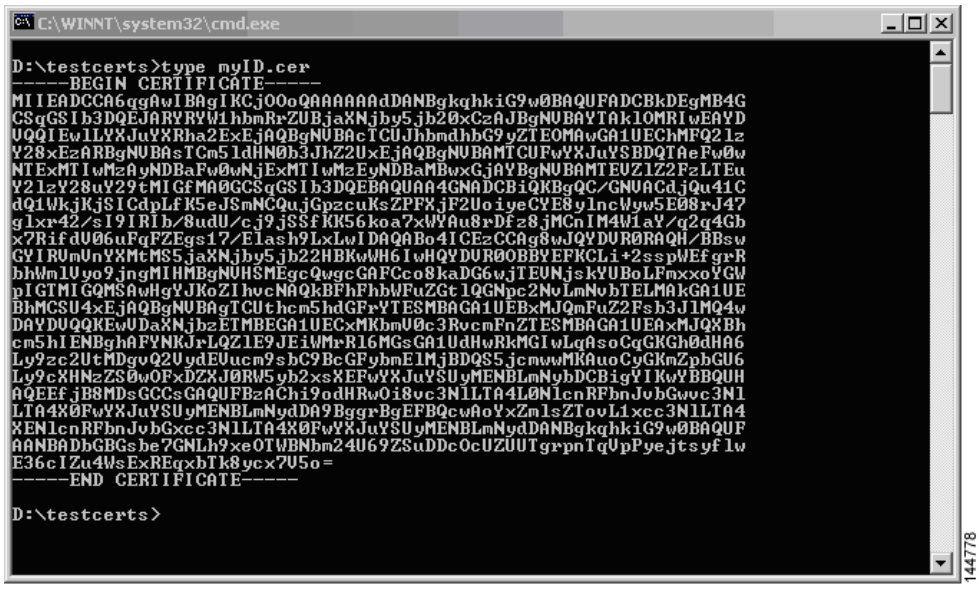

**Step 12** Enter the destination file name in the File name: text box on the Certificate Export Wizard dialog box, then click **Next**.




**Step 13** Click **Finish**.




**Step 14** Display the identity certificate in base64-encoded format using the Microsoft Windows **type** command.

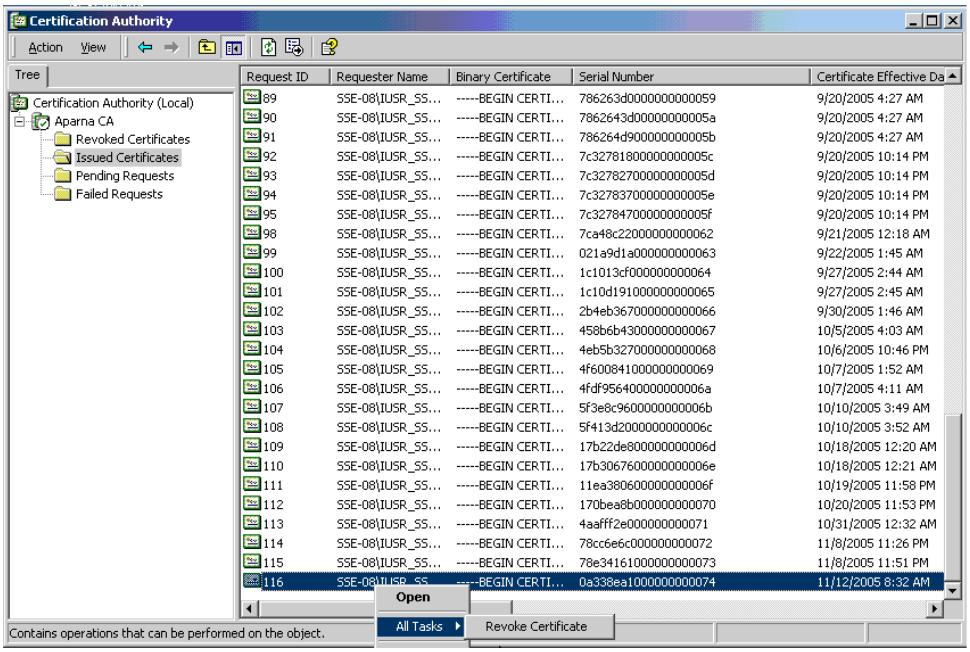



# Revoking a Certificate

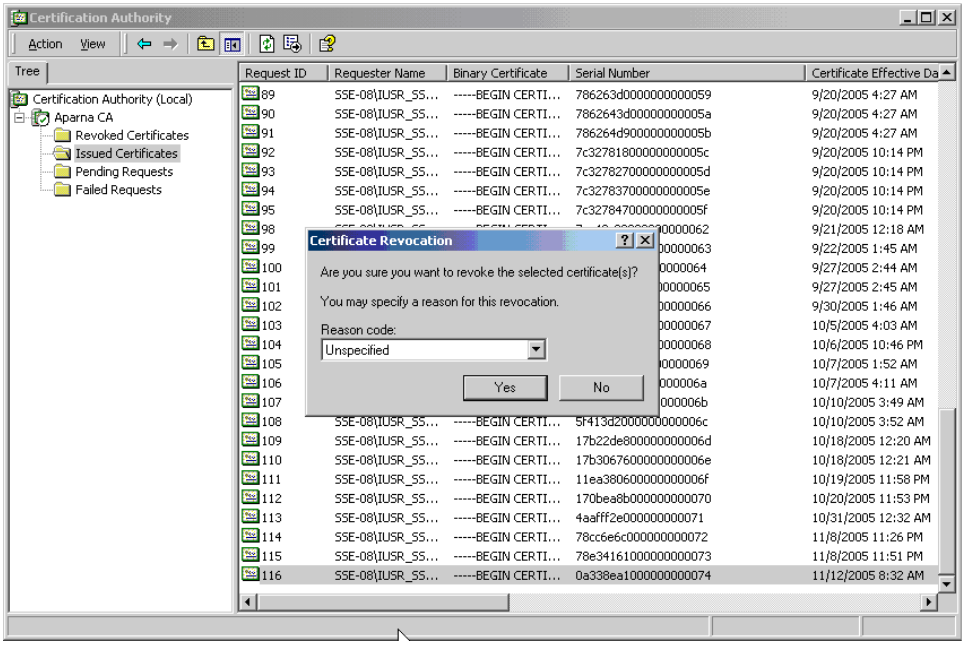
To revoke a certificate using the Microsoft CA administrator program, follow these steps:

**Step 1** Click the **Issued Certificates** folder on the Certification Authority tree. From the list, right-click the certificate you want to revoke.

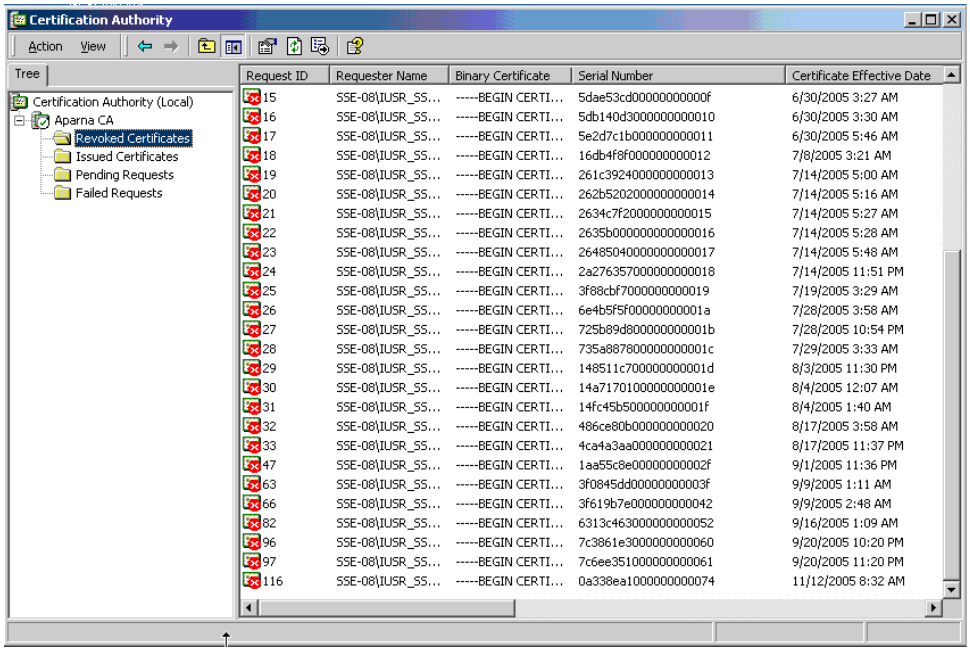
**Step 2** Select **All Tasks > Revoke Certificate**.




**Step 3** Select a reason for the revocation from the Reason code drop-down list, and click **Yes**.



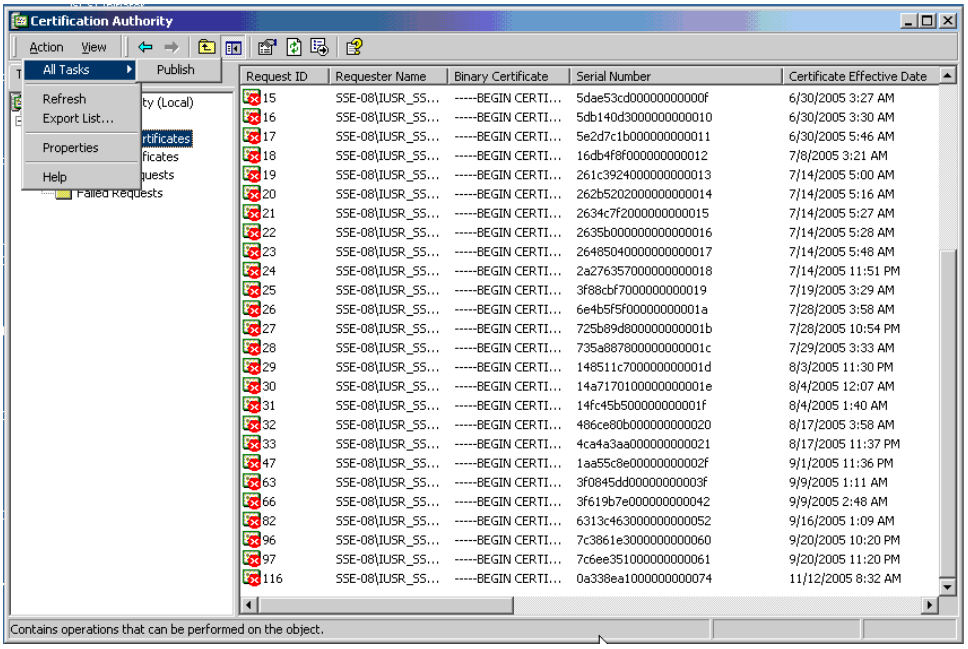

**Step 4** Click the **Revoked Certificates** folder to list and verify the certificate revocation.



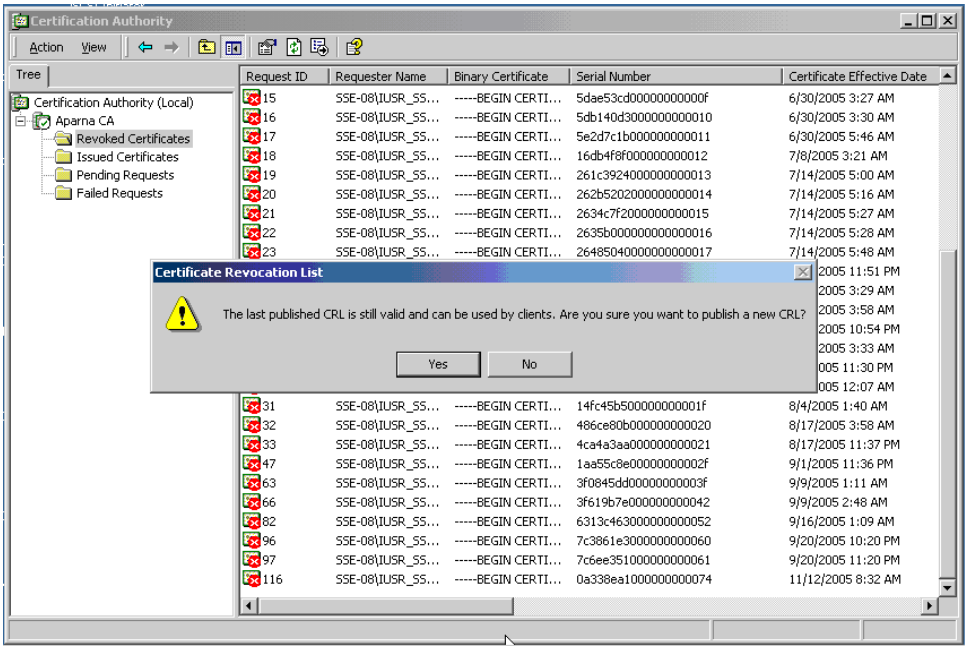

# Generating and Publishing the CRL

To generate and publish the CRL using the Microsoft CA administrator program, follow these steps:

**Step 1** Select **Action > All Tasks > Publish** on the Certification Authority screen.



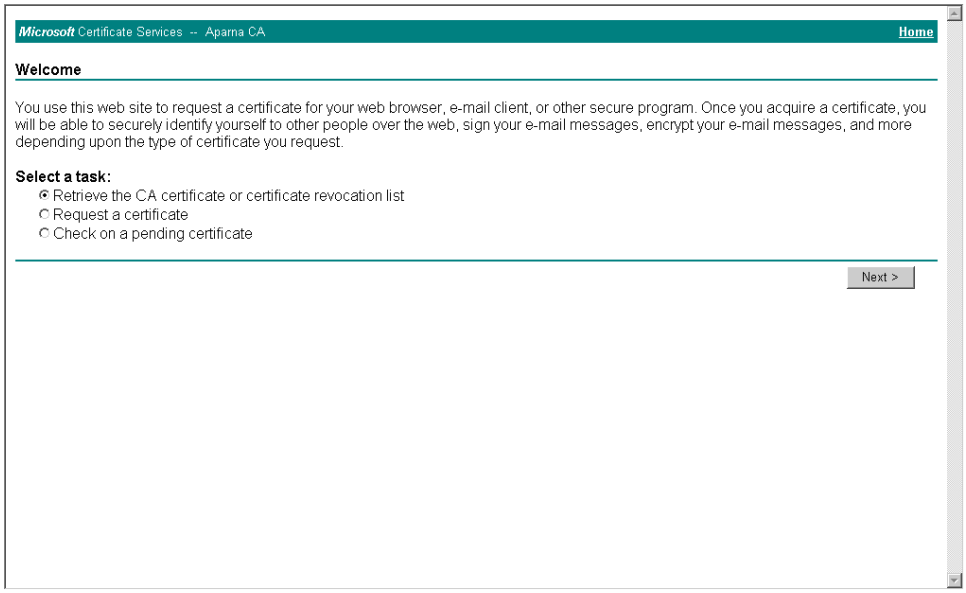

**Step 2** Click **Yes** on the Certificate Revocation List dialog box to publish the latest CRL.



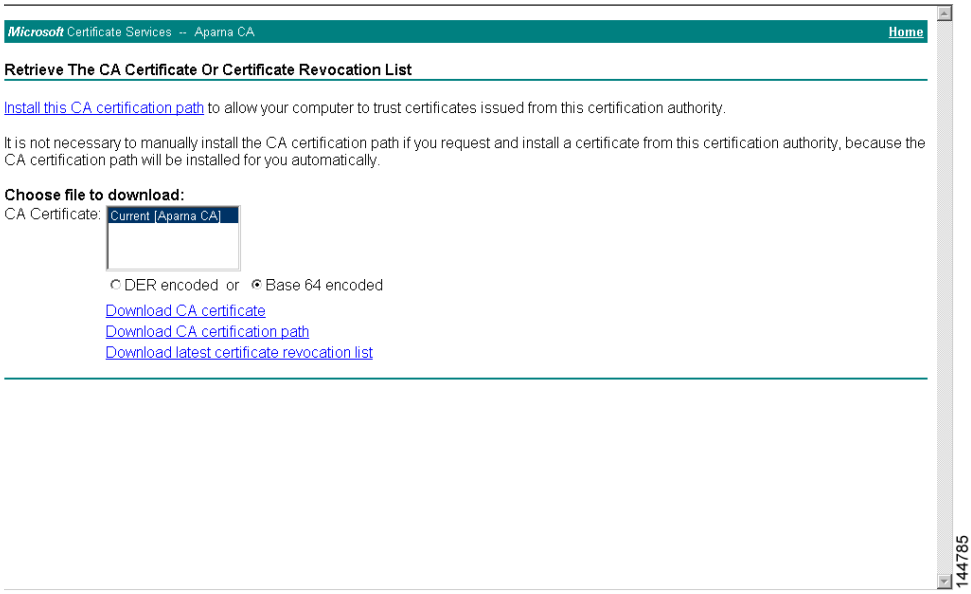

# Downloading the CRL

To download the CRL from the Microsoft CA website, follow these steps:

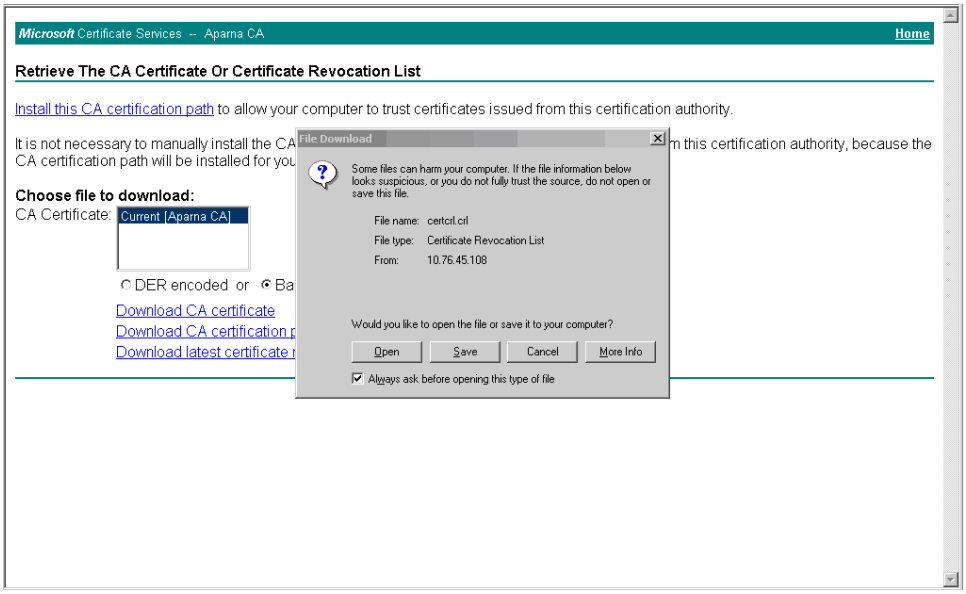
**Step 1** Click **Request the CA certificate or certificate revocation list** radio button on the Microsoft Certificate Services web interface and click **Next**.



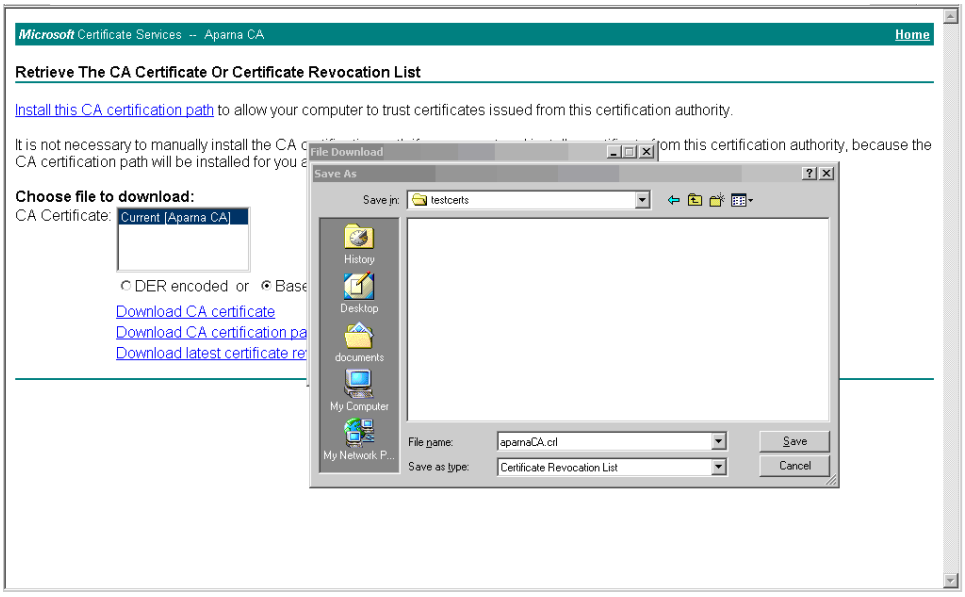

**Step 2** Click the **Download latest certificate revocation list** link.



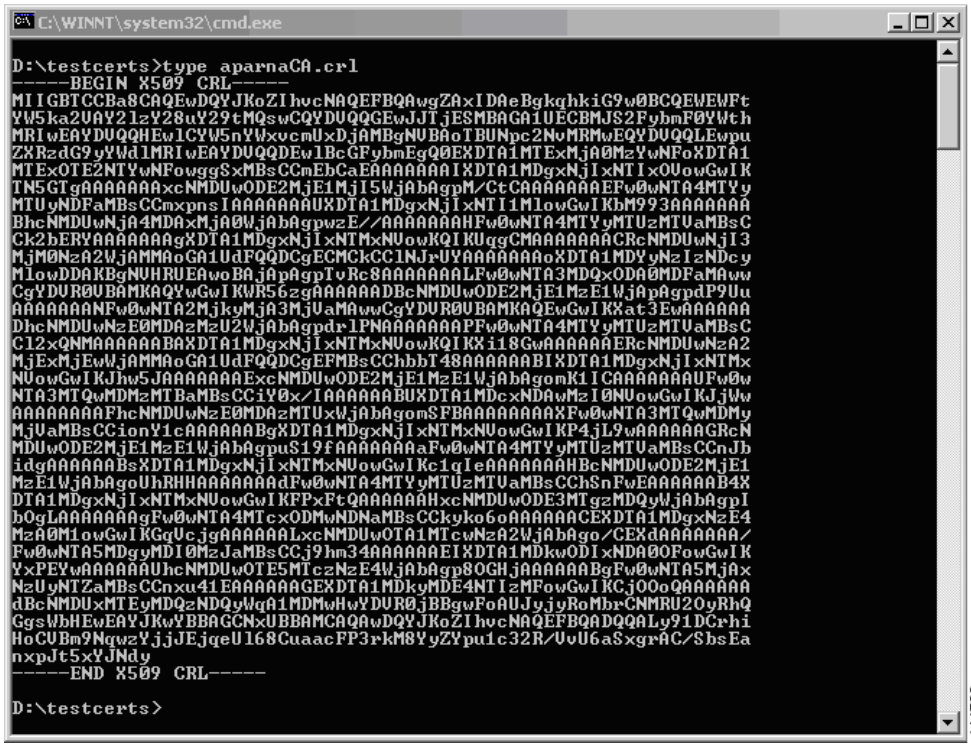

**Step 3** Click **Save** in the File Download dialog box.




**Step 4** Enter the destination file name in the Save As dialog box and click **Save**.




**Step 5** Display the CRL using the Microsoft Windows **type** command.




# Importing the CRL

To import the CRL to the trust point corresponding to the CA, follow these steps:

**Step 1** Copy the CRL file to the MDS switch bootflash.

```
Vegas-1# copy tftp:apranaCA.crl bootflash:aparnaCA.crl
```

**Step 2** Configure the CRL.

```
Vegas-1# config terminal
Vegas-1(config)# crypto ca crl request myCA bootflash:aparnaCA.crl
Vegas-1(config)#
```

**Step 3** Display the contents of the CRL.

```
Vegas-1(config)# show crypto ca crl myCA
Trustpoint: myCA
CRL:
Certificate Revocation List (CRL):
        Version 2 (0x1)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: /emailAddress=admin@yourcompany.com/C=IN/ST=Karnatak
Yourcompany/OU=netstorage/CN=Aparna CA
        Last Update: Nov 12 04:36:04 2005 GMT
        Next Update: Nov 19 16:56:04 2005 GMT
        CRL extensions:
            X509v3 Authority Key Identifier:
            keyid:27:28:F2:46:83:1B:AC:23:4C:45:4D:8E:C9:18:50:1
```

```
            1.3.6.1.4.1.311.21.1:
                ...
Revoked Certificates:
    Serial Number: 611B09A1000000000002
        Revocation Date: Aug 16 21:52:19 2005 GMT
Serial Number: 4CDE464E000000000003
        Revocation Date: Aug 16 21:52:29 2005 GMT
    Serial Number: 4CFC2B42000000000004
        Revocation Date: Aug 16 21:52:41 2005 GMT
    Serial Number: 6C699EC2000000000005
        Revocation Date: Aug 16 21:52:52 2005 GMT
    Serial Number: 6CCF7DDC000000000006
        Revocation Date: Jun  8 00:12:04 2005 GMT
    Serial Number: 70CC4FFF000000000007
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 4D9B1116000000000008
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 52A80230000000000009
        Revocation Date: Jun 27 23:47:06 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            CA Compromise
Serial Number: 5349AD4600000000000A
        Revocation Date: Jun 27 23:47:22 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            CA Compromise
Serial Number: 53BD173C00000000000B
        Revocation Date: Jul  4 18:04:01 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            Certificate Hold
Serial Number: 591E7ACE00000000000C
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 5D3FD52E00000000000D
        Revocation Date: Jun 29 22:07:25 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            Key Compromise
Serial Number: 5DAB771300000000000E
        Revocation Date: Jul 14 00:33:56 2005 GMT
    Serial Number: 5DAE53CD00000000000F
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 5DB140D3000000000010
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 5E2D7C1B000000000011
        Revocation Date: Jul  6 21:12:10 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            Cessation Of Operation
Serial Number: 16DB4F8F000000000012
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 261C3924000000000013
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 262B5202000000000014
        Revocation Date: Jul 14 00:33:10 2005 GMT
    Serial Number: 2634C7F2000000000015
        Revocation Date: Jul 14 00:32:45 2005 GMT
    Serial Number: 2635B000000000000016
        Revocation Date: Jul 14 00:31:51 2005 GMT
    Serial Number: 26485040000000000017
        Revocation Date: Jul 14 00:32:25 2005 GMT
    Serial Number: 2A276357000000000018
```

```
Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 3F88CBF7000000000019
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 6E4B5F5F00000000001A
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 725B89D800000000001B
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 735A8878000000000001C
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 148511C700000000001D
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 14A7170100000000001E
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 14FC45B500000000001F
        Revocation Date: Aug 17 18:30:42 2005 GMT
    Serial Number: 486CE80B000000000020
        Revocation Date: Aug 17 18:30:43 2005 GMT
    Serial Number: 4CA4A3AA000000000021
        Revocation Date: Aug 17 18:30:43 2005 GMT
    Serial Number: 1AA55C8E00000000002F
        Revocation Date: Sep  5 17:07:06 2005 GMT
    Serial Number: 3F0845DD00000000003F
        Revocation Date: Sep  8 20:24:32 2005 GMT
    Serial Number: 3F619B7E000000000042
        Revocation Date: Sep  8 21:40:48 2005 GMT
    Serial Number: 6313C463000000000052
        Revocation Date: Sep 19 17:37:18 2005 GMT
Serial Number: 7C3861E3000000000060
        Revocation Date: Sep 20 17:52:56 2005 GMT
    Serial Number: 7C6EE351000000000061
        Revocation Date: Sep 20 18:52:30 2005 GMT
    Serial Number: 0A338EA1000000000074       <-- Revoked identity certificate
        Revocation Date: Nov 12 04:34:42 2005 GMT
    Signature Algorithm: sha1WithRSAEncryption
        0b:cb:dd:43:0a:b8:62:1e:80:95:06:6f:4d:ab:0c:d8:8e:32:
        44:8e:a7:94:97:af:02:b9:a6:9c:14:fd:eb:90:cf:18:c9:96:
        29:bb:57:37:d9:1f:d5:bd:4e:9a:4b:18:2b:00:2f:d2:6e:c1:
        1a:9f:1a:49:b7:9c:58:24:d7:72
```

**Note** The identity certificate for the switch that was revoked (serial number 0A338EA1000000000074) is listed at the end.

# Maximum Limits

Table 6-1 lists the maximum limits for CAs and digital certificate parameters.

*Table 6-1 Maximum Limits for CA and Digital Certificate*

| Feature | Maximum Limit |
|---|---|
| Trust points declared on a switch | 16 |
| RSA key-pairs generated on a switch | 16 |
| Identity certificates configured on a switch | 16 |
| Certificates in a CA certificate chain | 10 |
| Trust points authenticated to a specific CA | 10 |

# Default Settings

Table 6-2 lists the default settings for CAs and digital certificate parameters.

*Table 6-2 Default CA and Digital Certificate Parameters*

| Parameters | Default |
|---|---|
| Trust point | None |
| RSA key-pair | None |
| RSA key-pair label | Switch FQDN |
| RSA key-pair modulus | 512 |
| RSA key-pair exportable | Yes |
| Revocation check method of trust point | CRL |


# Configuring IPsec Network Security

IP security (IPsec) protocol is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. It is developed by the Internet Engineering Task Force (IETF). IPsec provides security services at the IP layer, including protecting one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host. The overall IPsec implementation is the latest version of RFC 2401. Cisco NX-OS IPsec implements RFC 2402 through RFC 2410.

IPsec uses the Internet Key Exchange (IKE) protocol to handle protocol and algorithm negotiation and to generate the encryption and authentication keys used by IPsec. While IKE can be used with other protocols, its initial implementation is with the IPsec protocol. IKE provides authentication of the IPsec peers, negotiates IPsec security associations, and establishes IPsec keys. IKE uses RFCs 2408, 2409, 2410, and 2412, and additionally implements the draft-ietf-ipsec-ikev2-16.txt draft.

**Note** The term IPsec is sometimes used to describe the entire protocol of IPsec data services and IKE security protocols and is other times used to describe only the data services.

This chapter includes the following sections:

• Default Settings, page 7-209

# Feature Information

This section briefly describes the new and updated features for releases.

*Table 7-1 New and Changed Features*

| Feature | Release | Description |
|---|---|---|
| SHA2 support for IPSec and IKEv2 on Cisco MDS 9700 Series Switches | 7.3(1)DY(1) | This feature enables SHA2 support for IPSec and IKEv2 on Cisco MDS 9700 Series Switches |
| SHA2 support for IPSec and IKEv2 | 7.3(0)D1(1) | This feature enables SHA2 support for IPSec and IKEv2 on a Cisco MDS 9250i Switch. |

# About IPsec

IP security (IPsec) protocol is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. It is developed by the Internet Engineering Task Force (IETF). IPsec provides security services at the IP layer, including protecting one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host. The overall IPsec implementation is the latest version of RFC 2401. Cisco NX-OS IPsec implements RFC 2402 through RFC 2410.

IPsec uses the Internet Key Exchange (IKE) protocol to handle protocol and algorithm negotiation and to generate the encryption and authentication keys used by IPsec. While IKE can be used with other protocols, its initial implementation is with the IPsec protocol. IKE provides authentication of the IPsec peers, negotiates IPsec security associations, and establishes IPsec keys. IKE uses RFCs 2408, 2409, 2410, and 2412, and additionally implements the draft-ietf-ipsec-ikev2-16.txt draft.

IPsec provides security for transmission of sensitive information over unprotected networks such as the Internet. IPsec acts at the network layer, protecting and authenticating IP packets between participating IPsec devices (peers).

**Note** IPsec is not supported by the Cisco Fabric Switch for HP c-Class BladeSystem and the Cisco Fabric Switch for IBM BladeCenter.

IPsec provides security for transmission of sensitive information over unprotected networks such as the Internet. IPsec acts at the network layer, protecting and authenticating IP packets between participating IPsec devices (peers).

IPsec provides the following network security services. In general, the local security policy dictates the use of one or more of these services between two participating IPsec devices:

- Data confidentiality—The IPsec sender can encrypt packets before transmitting them across a network.
- Data integrity—The IPsec receiver can authenticate packets sent by the IPsec sender to ensure that the data has not been altered during transmission.

- Data origin authentication—The IPsec receiver can authenticate the source of the IPsec packets sent. This service is dependent upon the data integrity service.
- Anti-replay protection—The IPsec receiver can detect and reject replayed packets.

**Note** The term *data authentication* is generally used to mean data integrity and data origin authentication. Within this chapter it also includes anti-replay services, unless otherwise specified.

With IPsec, data can be transmitted across a public network without fear of observation, modification, or spoofing. This enables applications such as Virtual Private Networks (VPNs), including intranets, extranets, and remote user access.

IPsec as implemented in Cisco NX-OS software supports the Encapsulating Security Payload (ESP) protocol. This protocol encapsulates the data to be protected and provides data privacy services, optional data authentication, and optional anti-replay services.

**Note** The Encapsulating Security Payload (ESP) protocol is a header inserted into an existing TCP/IP packet, the size of which depends on the actual encryption and authentication algorithms negotiated. To avoid fragmentation, the encrypted packet fits into the interface maximum transmission unit (MTU). The path MTU calculation for TCP takes into account the addition of ESP headers, plus the outer IP header in tunnel mode, for encryption. The MDS switches allow 100 bytes for packet growth for IPsec encryption.

**Note** When using IPsec and IKE, each Gigabit Ethernet interface on the IPS module (on 18+4, and 24/10 port SAN Extension modules) must be configured in its own IP subnet. If there are multiple Gigabit Ethernet interfaces configured with IP address or network-mask in the same IP subnet, IKE packets may not be sent to the right peer and thus IPsec tunnel will not come up.

# About IKE

IKE automatically negotiates IPsec security associations and generates keys for all switches using the IPsec feature. Specifically, IKE provides these benefits:

- Allows you to refresh IPsec SAs.
- Allows IPsec to provide anti-replay services.
- Supports a manageable, scalable IPsec configuration.
- Allows dynamic authentication of peers.

**Note** IKE is not supported on the Cisco Fabric Switch for HP c-Class BladeSystem and the Cisco Fabric Switch for IBM BladeSystem.

# IPsec Prerequisites

To use the IPsec feature, you need to perform the following tasks:

- Obtain the ENTERPRISE_PKG license (see the *Cisco MDS 9000 Family NX-OS Licensing Guide*).

- Configure IKE as described in the "About IKE Initialization" section on page 7-180.

# Using IPsec

To use the IPsec feature, follow these steps:

**Step 1** Obtain the ENTERPRISE_PKG license to enable IPsec for iSCSI and to enable IPsec for FCIP. See the *Cisco MDS 9000 Family NX-OS Licensing Guide* for more information.

**Step 2** Configure IKE as described in the "Manually Configuring IPsec and IKE" section on page 7-180.

**Note** The IPsec feature inserts new headers in existing packets (see the Cisco MDS 9000 Family NX-OS IP Services Configuration Guide for more information).

This section contains the following topics:



## IPsec Compatibility

IPsec features are compatible with the following Cisco MDS 9000 Family hardware:

- Cisco 18/4-port Multi-Service Module (MSM-18/4).
- Cisco MDS 9250i Multiservice Fabric Switches.
- Cisco MDS 24/10 port SAN Extension Module on Cisco MDS 9700 Series Switches.
- The IPsec feature is not supported on the management interface.

IPsec features are compatible with the following fabric setup:

- Two connected Cisco MDS 9200 Switches or Cisco MDS 9500 Directors running Cisco MDS SAN-OS Release 2.0(1b) or later, or Cisco NX-OS 4.1(1).
- A Cisco MDS 9200 Switches or Cisco MDS 9500 Directors running Cisco MDS SAN-OS Release 2.0(1b) or later, or Cisco NX-OS 4.1(1) connected to any IPsec compliant device.
- The following features are not supported in the Cisco NX-OS implementation of the IPsec feature:
  - Authentication Header (AH).
  - Transport mode.
  - Security association bundling.
  - Manually configuring security associations.
  - Per host security association option in a crypto map.
  - Security association idle timeout

  - Dynamic crypto maps.

**Note** Any reference to crypto maps in this document, only refers to static crypto maps.

# IPsec and IKE Terminology

The terms used in this chapter are explained in this section.

- Security association (SA)— An agreement between two participating peers on the entries required to encrypt and decrypt IP packets. Two SAs are required for each peer in each direction (inbound and outbound) to establish bidirectional communication between the peers. Sets of bidirectional SA records are stored in the SA database (SAD). IPsec uses IKE to negotiate and bring up SAs. Each SA record includes the following information:
  - Security parameter index (SPI)—A number which, together with a destination IP address and security protocol, uniquely identifies a particular SA. When using IKE to establish the SAs, the SPI for each SA is a pseudo-randomly derived number.
  - Peer—A switch or other device that participates in IPsec. For example, a Cisco MDS switch or other Cisco routers that support IPsec.
  - Transform—A list of operations done to provide data authentication and data confidentiality. For example, one transform is the ESP protocol with the HMAC-MD5 authentication algorithm.
  - Session key—The key used by the transform to provide security services.
  - Lifetime—A lifetime counter (in seconds and bytes) is maintained from the time the SA is created. When the time limit expires the SA is no longer operational and, if required, is automatically renegotiated (rekeyed).
  - Mode of operation—Two modes of operation are generally available for IPsec: tunnel mode and transport mode. The Cisco NX-OS implementation of IPsec only supports the tunnel mode. The IPsec tunnel mode encrypts and authenticates the IP packet, including its header. The gateways encrypt traffic on behalf of the hosts and subnets.
    The Cisco NX-OS implementation of IPsec does not support transport mode.

  **Note** The term *tunnel mode* is different from the term *tunnel*, which is used to indicate a secure communication path between two peers, such as two switches connected by an FCIP link.

- Anti-replay—A security service where the receiver can reject old or duplicate packets to protect itself against replay attacks. IPsec provides this optional service by use of a sequence number combined with the use of data authentication.
- Data authentication—Data authentication can refer either to integrity alone or to both integrity and authentication (data origin authentication is dependent on data integrity).
  - Data integrity—Verifies that data has not been altered.
  - Data origin authentication—Verifies that the data was actually sent by the claimed sender.
- Data confidentiality—A security service where the protected data cannot be observed.
- Data flow—A grouping of traffic, identified by a combination of source address and mask or prefix, destination address mask or prefix length, IP next protocol field, and source and destination ports, where the protocol and port fields can have any of these values. Traffic matching a specific

combination of these values is logically grouped together into a data flow. A data flow can represent a single TCP connection between two hosts, or it can represent traffic between two subnets. IPsec protection is applied to data flows.

- Perfect forward secrecy (PFS)—A cryptographic characteristic associated with a derived shared secret value. With PFS, if one key is compromised, previous and subsequent keys are not compromised, because subsequent keys are not derived from previous keys.
- Security Policy Database (SPD)—An ordered list of policies applied to traffic. A policy decides if a packet requires IPsec processing, if it should be allowed in clear text, or if it should be dropped.
  - The IPsec SPDs are derived from user configuration of crypto maps.
  - The IKE SPD is configured by the user.

# Supported IPsec Transforms and Algorithms

The component technologies implemented for IPsec include the following transforms:

- Advanced Encrypted Standard (AES) is an encryption algorithm. It implements either 128 or 256 bits using Cipher Block Chaining (CBC) or counter mode.
- Data Encryption Standard (DES) is used to encrypt packet data and implements the mandatory 56-bit DES-CBC. CBC requires an initialization vector (IV) to start encryption. The IV is explicitly given in the IPsec packet.
- Triple DES (3DES) is a stronger form of DES with 168-bit encryption keys that allow sensitive information to be transmitted over untrusted networks.

**Note** Cisco NX-OS images with strong encryption are subject to United States government export controls, and have a limited distribution. Images to be installed outside the United States require an export license. Customer orders might be denied or subject to delay due to United States government regulations. Contact your sales representative or distributor for more information, or send e-mail to export@cisco.com.

- Message Digest 5 (MD5) is a hash algorithm with the HMAC variant. HMAC is a keyed hash variant used to authenticate data.
- Secure Hash Algorithm (SHA-1, SHA-2) is a hash algorithm with the Hash Message Authentication Code (HMAC) variant. IPsec supports SHA-2 on Cisco MDS 9250i Multiservice Fabric Switches starting from Cisco MDS NX-OS Release 7.3(0)D1(1).
- AES-XCBC-MAC is a Message Authentication Code (MAC) using the AES algorithm.
- IPsec supports SHA-2 on Cisco MDS 24/10 port SAN Extension Modules (Cisco MDS 9700 Series Switches) starting from Cisco MDS NX-OS Release 7.3(1)DY(1).

# Supported IKE Transforms and Algorithms

The component technologies implemented for IKE include the following transforms:

- Diffie-Hellman (DH) is a public-key cryptography protocol that allows two parties to establish a shared secret over an unsecure communications channel. Diffie-Hellman is used within IKE to establish session keys. Group 1 (768-bit), Group 2 (1024-bit), and Group 5 (1536-bit) are supported.

- Advanced Encrypted Standard (AES) is an encryption algorithm. It implements either 128 bits using Cipher Block Chaining (CBC) or counter mode.
- Data Encryption Standard (DES) is used to encrypt packet data and implements the mandatory 56-bit DES-CBC. CBC requires an initialization vector (IV) to start encryption. The IV is explicitly given in the IPsec packet.
- Triple DES (3DES) is a stronger form of DES with 168-bit encryption keys that allow sensitive information to be transmitted over untrusted networks.

**Note** Cisco NX-OS images with strong encryption are subject to United States government export controls, and have a limited distribution. Images to be installed outside the United States require an export license. Customer orders might be denied or subject to delay due to United States government regulations. Contact your sales representative or distributor for more information, or send e-mail to export@cisco.com.

- Message Digest 5 (MD5) is a hash algorithm with the HMAC variant. HMAC is a keyed hash variant used to authenticate data.
- Secure Hash Algorithm (SHA-1, SHA-2) is a hash algorithm with the Hash Message Authentication Code (HMAC) variant. IKEv2 supports SHA-2 on Cisco MDS 9250i Multiservice Fabric Switches starting from Cisco MDS NX-OS Release 7.3(0)D1(1).

**Note** IKEv1 does not support SHA-2.

- The switch authentication algorithm uses the preshared keys based on the IP address.
- IKEv2 supports SHA-2 on Cisco MDS 24/10 port SAN Extension Modules (Cisco MDS 9700 Series Switches) starting from Cisco MDS NX-OS Release 7.3(1)DY(1).

# IPsec Digital Certificate Support

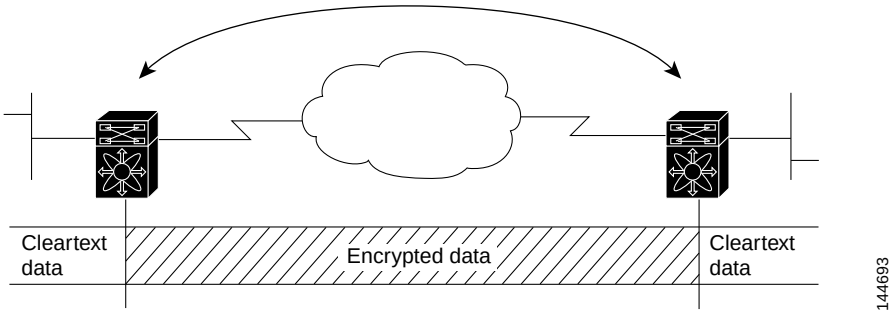
This section describes the advantages of using certificate authorities (CAs) and digital certificates for authentication.

## Implementing IPsec Without CAs and Digital Certificates

Without a CA and digital certificates, enabling IPsec services (such as encryption) between two Cisco MDS switches requires that each switch has the key of the other switch (such as an RSA public key or a shared key). You must manually specify either the RSA public keys or preshared keys on each switch in the fabric using IPsec services. Also, each new device added to the fabric will require manual configuration of the other switches in the fabric to support secure communication. Each (see Figure 7-1) switch uses the key of the other switch to authenticate the identity of the other switch; this authentication always occurs when IPsec traffic is exchanged between the two switches.

If you have multiple Cisco MDS switches in a mesh topology and wish to exchange IPsec traffic passing among all of those switches, you must first configure shared keys or RSA public keys among all of those switches.

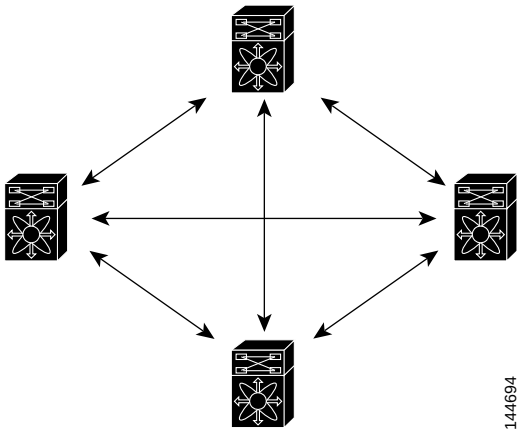
*Figure 7-1 Two IPsec Switches Without CAs and Digital Certificates*




Every time a new switch is added to the IPsec network, you must configure keys between the new switch and each of the existing switches. (In Figure 7-2, four additional two-part key configurations are required to add a single encrypting switch to the network).

Consequently, the more devices that require IPsec services, the more involved the key administration becomes. This approach does not scale well for larger, more complex encrypting networks.

*Figure 7-2 Four IPsec Switches Without a CA and Digital Certificates*

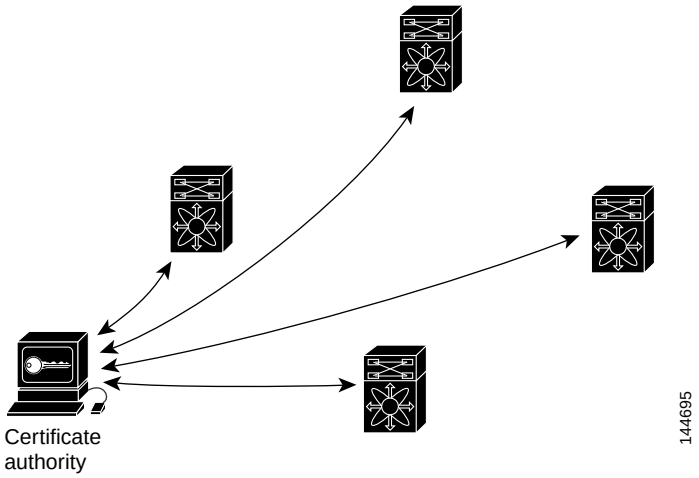



# Implementing IPsec with CAs and Digital Certificates

With CA and digital certificates, you do not have to configure keys between all the encrypting switches. Instead, you individually enroll each participating switch with the CA, requesting a certificate for the switch. When this has been accomplished, each participating switch can dynamically authenticate all the other participating switches. When two devices want to communicate, they exchange certificates and digitally sign data to authenticate each other. When a new device is added to the network, you simply enroll that device with a CA, and none of the other devices needs modification. When the new device attempts an IPsec connection, certificates are automatically exchanged and the device can be authenticated.

Figure 7-3 shows the process of dynamically authenticating the devices.

*Figure 7-3 Dynamically Authenticating Devices with a CA*




To add a new IPsec switch to the network, you need only configure that new switch to request a certificate from the CA, instead of making multiple key configurations with all the other existing IPsec switches.

# How CA Certificates Are Used by IPsec Devices

When two IPsec switches want to exchange IPsec-protected traffic passing between them, they must first authenticate each other—otherwise, IPsec protection cannot occur. The authentication is done with IKE.

IKE can use two methods to authenticate the switches, using preshared keys without a CA and using RSA key-pairs with a CA. Both methods require that keys must be preconfigured between the two switches.

Without a CA, a switch authenticates itself to the remote switch using either RSA-encrypted preshared keys.

With a CA, a switch authenticates itself to the remote switch by sending a certificate to the remote switch and performing some public key cryptography. Each switch must send its own unique certificate that was issued and validated by the CA. This process works because the certificate of each switch encapsulates the public key of the switch, each certificate is authenticated by the CA, and all participating switches recognize the CA as an authenticating authority. This scheme is called IKE with an RSA signature.

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

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

Certificate support for IKE has the following considerations:

- The switch FQDN (host name and domain name) must be configured before installing certificates for IKE.
- Only those certificates that are configured for IKE or general usage are used by IKE.
- The first IKE or general usage certificate configured on the switch is used as the default certificate by IKE.
- The default certificate is for all IKE peers unless the peer specifies another certificate.

- If the peer asks for a certificate which is signed by a CA that it trusts, then IKE uses that certificate, if it exists on the switch, even if it is not the default certificate.
- If the default certificate is deleted, the next IKE or general usage certificate, if any exists, is used by IKE as the default certificate.
- Certificate chaining is not supported by IKE.
- IKE only sends the identity certificate, not the entire CA chain. For the certificate to be verified on the peer, the same CA chain must also exist there.

# Manually Configuring IPsec and IKE

This section describes how to manually configure IPsec and IKE.

IPsec provides secure data flows between participating peers. Multiple IPsec data flows can exist between two peers to secure different data flows, with each tunnel using a separate set of SAs.

After you have completed IKE configuration, configure IPsec.

To configure IPsec in each participating IPsec peer, follow these steps:

**Step 1** Identify the peers for the traffic to which secure tunnels should be established.

**Step 2** Configure the transform set with the required protocols and algorithms.

**Step 3** Create the crypto map and apply access control lists (IPv4-ACLs), transform sets, peers, and lifetime values as applicable.

**Step 4** Apply the crypto map to the required interface.

This section contains the following topics:



## About IKE Initialization

The IKE feature must first be enabled and configured so the IPsec feature can establish data flow with the required peer. Fabric Manager initializes IKE when you first configure it.

You cannot disable IKE if IPsec is enabled. If you disable the IKE feature, the IKE configuration is cleared from the running configuration.

# Enabling IKE

To enable IKE, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config terminal` | Enters configuration mode. |
| **Step 2** | `switch(config)# feature crypto ike` | Enables the IKE feature. |
| | `switch(config)# no feature crypto ike` | Disables (default) the IKE feature.<br>**Note** You must disable IPsec before you can disable the IKE feature. |

# About the IKE Domain

You must apply the IKE configuration to an IPsec domain to allow traffic to reach the supervisor module in the local switch. Fabric Manager sets the IPsec domain automatically when you configure IKE.

# Configuring the IKE Domain

You must apply the IKE configurations to an IPsec domain to allow traffic to reach the supervisor module in the local switch.

To configure the IPsec domain, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config terminal` | Enters configuration mode. |
| **Step 2** | `switch(config)# crypto ike domain ipsec` | Allows IKE configurations for IPsec domains. |

# About IKE Tunnels

An IKE tunnel is a secure IKE session between two endpoints. IKE creates this tunnel to protect IKE messages used in IPsec SA negotiations.

Two versions of IKE are used in the Cisco NX-OS implementation.

- IKE version 1 (IKEv1) is implemented using RFC 2407, 2408, 2409, and 2412.
- IKE version 2 (IKEv2) is a simplified and more efficient version and does not interoperate with IKEv1. IKEv2 is implemented using the draft-ietf-ipsec-ikev2-16.txt draft.

# About IKE Policy Negotiation

To protect IKE negotiations, each IKE negotiation begins with a common (shared) IKE policy. An IKE policy defines a combination of security parameters to be used during the IKE negotiation. By default, no IKE policy is configured. You must create IKE policies at each peer. This policy states which security parameters will be used to protect subsequent IKE negotiations and mandates how peers are authenticated. You can create multiple, prioritized policies at each peer to ensure that at least one policy will match a remote peer's policy.

You can configure the policy based on the encryption algorithm (DES, 3DES, or AES), the hash algorithm (SHA or MD5), and the DH group (1, 2, or 5). Each policy can contain a different combination of parameter values. A unique priority number identifies the configured policy. This number ranges from 1 (highest priority) to 255 (lowest priority). You can create multiple policies in a switch. If you need to connect to a remote peer, you must ascertain that at least one policy in the local switch contains the identical parameter values configured in the remote peer. If several policies have identical parameter configurations, the policy with the lowest number is selected.

Table 7-2 provides a list of allowed transform combinations.

*Table 7-2 IKE Transform Configuration Parameters*

| Parameter | Accepted Values | Keyword | Default Value |
|---|---|---|---|
| encryption algorithm | 56-bit DES-CBC | **des** | **3des** |
| | 168-bit DES | **3des** | |
| | 128-bit AES | **aes** | |
| hash algorithm | SHA-1 (HMAC variant) | **sha** | **sha** |
| | SHA-2 (HMAC variant) | **sha256** | |
| | MD5 (HMAC variant) | **sha512** | |
| | | **md5** | |
| authentication method | Preshared keys | Not configurable | Preshared keys |
| DH group identifier | 768-bit DH | **1** | **1** |
| | 1024-bit DH | **2** | |
| | 1536-bit DH | **5** | |

The following table lists the supported and verified settings for IPsec and IKE encryption authentication algorithms on the Microsoft Windows and Linux platforms:

| Platform | IKE | IPsec |
|---|---|---|
| Microsoft iSCSI initiator, Microsoft IPsec implementation on Microsoft Windows 2000 platform | 3DES, SHA-1, SHA-2, or MD5, DH group 2 | 3DES, SHA-1, SHA-2 |
| Cisco iSCSI initiator, Free Swan IPsec implementation on Linux platform | 3DES, MD5, DH group 1 | 3DES, MD5 |

**Note** When you configure the hash algorithm, the corresponding HMAC version is used as the authentication algorithm.

When the IKE negotiation begins, IKE looks for an IKE policy that is the same on both peers. The peer that initiates the negotiation will send all its policies to the remote peer, and the remote peer will try to find a match. The remote peer looks for a match by comparing its own highest priority policy against the other peer's received policies. The remote peer checks each of its policies in order of its priority (highest priority first) until a match is found.

A match is found when the two peers have the same encryption, hash algorithm, authentication algorithm, and DH group values. If a match is found, IKE completes the security negotiation and the IPsec SAs are created.

If an acceptable match is not found, IKE refuses negotiation and the IPsec data flows will not be established.

# Configuring an IKE Policy

To configure the IKE negotiation parameters, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config terminal` | Enters configuration mode. |
| **Step 2** | `switch(config)# crypto ike domain ipsec` | Allows IPsec domains to be configured in this switch. |
| **Step 3** | `switch(config-ike-ipsec)# identity address` | Configures the identity mode for the IKE protocol to use the IP address (default). |
| | `switch(config-ike-ipsec)# identity hostname` | Configures the identity mode for the IKE protocol to use the fully-qualified domain name (FQDN).<br>**Note** The FQDN is required for using RSA signatures for authentication. |
| **Step 4** | `switch(config-ike-ipsec)# key switch1 address 10.10.1.1` | Associates a preshared key with the IP address of a peer. |
| | `switch(config-ike-ipsec)# key switch1 hostname switch1.cisco.com` | Associates a preshared key with the FQDN of a peer.<br>**Note** To use the FQDN, you must configure the switch name and domain name on the peer. |
| **Step 5** | `switch(config-ike-ipsec)# policy 1` | Specifies the policy to configure. |
| **Step 6** | `switch(config-ike-ipsec-policy)# encryption des` | Configures the encryption policy. |
| **Step 7** | `switch(config-ike-ipsec-policy)# group 5` | Configures the DH group. |
| **Step 8** | `switch(config-ike-ipsec-policy)# hash md5` | Configures the hash algorithm. |
| **Step 9** | `switch(config-ike-ipsec-policy)# authentication pre-share` | Configures the authentication method to use the preshared key (default). |
| | `switch(config-ike-ipsec-policy)# authentication rsa-sig` | Configures the authentication method to use the RSA signature.<br>**Note** To use RSA signatures for authentication you must configure identity authentication mode using the FQDN (see Step 3). |

**Note** When the authentication method is rsa-sig, make sure the identity hostname is configured for IKE because the IKE certificate has a subject name of the FQDN type.

**Note** Before you downgrade to Cisco MDS NX-OS Release 5.2(x), unconfigure the preshared key. Once downgrading is complete, reconfigure the preshared key using the **key** *key-name* **hostname** *host* or **key** *key-name* **address** *ip-address* commands.

# Optional IKE Parameter Configuration

You can optionally configure the following parameters for the IKE feature:

- The lifetime association within each policy—The lifetime ranges from 600 to 86,400 seconds. The default is 86,400 seconds (equals one day). The lifetime association within each policy is configured when you are creating an IKE policy. See the "Configuring an IKE Policy" section on page 7-183.
- The keepalive time for each peer if you use IKEv2—The keepalive ranges from 120 to 86,400 seconds. The default is 3,600 seconds (equals one hour).
- The initiator version for each peer—IKE v1 or IKE v2 (default). Your choice of initiator version does not affect interoperability when the remote device initiates the negotiation. Configure this option if the peer device supports IKEv1 and you can play the initiator role for IKE with the specified device. Use the following considerations when configuring the initiator version with FCIP tunnels:
  - If the switches on both sides of an FCIP tunnel are running MDS SAN-OS Release 3.0(1) or later, or Cisco NX-OS 4.1(1) you must configure initiator version IKEv1 on both sides of an FCIP tunnel to use only IKEv1. If one side of an FCIP tunnel is using IKEv1 and the other side is using IKEv2, the FCIP tunnel uses IKEv2.
  - If the switch on one side of an FCIP tunnel is running MDS SAN-OS Release 3.0(1) or later, or Cisco NX-OS 4.1(1b) and the switch on the other side of the FCIP tunnel is running MDS SAN-OS Release 2.x, configuring IKEv1 on either side (or both) results in the FCIP tunnel using IKEv1.

**Note** Only IKE v1 is supported to build IPsec between 2.x and 3.x MDS switches.

**Caution** You may need to configure the initiator version even when the switch does not behave as an IKE initiator under normal circumstances. Always using this option guarantees a faster recovery of traffic flows in case of failures.

**Tip** The keepalive time only applies to IKEv2 peers and not to all peers.

**Note** When IPsec implementations in the host prefer to initiate the IPsec rekey, be sure to configure the IPsec lifetime value in the Cisco MDS switch to be higher than the lifetime value in the host.

This section includes the following topics:

# Configuring the Lifetime Association for a Policy

To configure the lifetime association for each policy, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config terminal` | Enters configuration mode. |
| **Step 2** | `switch(config)# crypto ike domain ipsec` | Allows IPsec domains to be configured in this switch. |
| **Step 3** | `switch(config-ike-ipsec)# policy 1` | Specifies the policy to configure. |
| **Step 4** | `switch(config-ike-ipsec-policy) lifetime seconds 6000` | Configures a lifetime of 6,000 seconds. |

# Configuring the Keepalive Time for a Peer

To configure the keepalive time for each peer, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config terminal` | Enters configuration mode. |
| **Step 2** | `switch(config)# crypto ike domain ipsec` | Allows IPsec domains to be configured in this switch. |
| **Step 3** | `switch(config-ike-ipsec)# keepalive 60000` | Configures the keepalive time for all peers to be 60,000 seconds. |

# Configuring the Initiator Version

To configure the initiator version using IPv4, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config terminal` | Enters configuration mode. |
| **Step 2** | `switch(config)# crypto ike domain ipsec` | Allows IPsec domains to be configured in this switch. |
| **Step 3** | `switch(config-ike-ipsec)# initiator version 1 address 10.10.10.1` | Configures the switch to use IKEv1 when initiating IKE with device 10.10.10.0<br>**Note** IKE supports IPv4 addresses, not IPv6 addresses. |

# Clearing IKE Tunnels or Domains

If an IKE tunnel ID is not specified for the IKE configuration, you can clear all existing IKE domain connections by issuing the **clear crypto ike domain ipsec sa** command in EXEC mode.

```
switch# clear crypto ike domain ipsec sa
```

**Caution** When you delete all the SAs within a specific IKEv2 tunnel, then that IKE tunnel is automatically deleted.

If an SA is specified for the IKE configuration, you can clear the specified IKE tunnel ID connection by issuing the **clear crypto ike domain ipsec sa** *IKE_tunnel-ID* command in EXEC mode.

```
switch# clear crypto ike domain ipsec sa 51
```

**Caution** When you delete the IKEv2 tunnel, the associated IPsec tunnel under that IKE tunnel is automatically deleted.

## Refreshing SAs

Use the **crypto ike domain ipsec rekey IPv4-ACL**-*index* command to refresh the SAs after performing IKEv2 configuration changes.

# Crypto IPv4-ACLs

IP access control lists (IPv4-ACLs) provide basic network security to all switches in the Cisco MDS 9000 Family. IPv4 IP-ACLs restrict IP-related traffic based on the configured IP filters. See Chapter 5, "Configuring IPv4 and IPv6 Access Control Lists" for details on creating and defining IPv4-ACLs.

In the context of crypto maps, IPv4-ACLs are different from regular IPv4-ACLs. Regular IPv4-ACLs determine what traffic to forward or block at an interface. For example, IPv4-ACLs can be created to protect all IP traffic between subnet A and subnet Y or Telnet traffic between host A and host B.

This section contains the following topics:

# About Crypto IPv4-ACLs

Crypto IPv4-ACLs are used to define which IP traffic requires crypto protection and which traffic does not.

Crypto IPv4-ACLs associated with IPsec crypto map entries have four primary functions:

- Select outbound traffic to be protected by IPsec (permit = protect).
- Indicate the data flow to be protected by the new SAs (specified by a single permit entry) when initiating negotiations for IPsec SAs.
- Process inbound traffic to filter out and discard traffic that should have been protected by IPsec.
- Determine whether or not to accept requests for IPsec SAs on behalf of the requested data flows when processing IKE negotiation from the IPsec peer.

**Tip** If you want some traffic to receive one type of IPsec protection (for example, encryption only) and other traffic to receive a different type of IPsec protection (for example, both authentication and encryption), create two IPv4-ACLs. Use both IPv4-ACLs in different crypto maps to specify different IPsec policies.

**Note** IPsec does not support IPv6-ACLs.

## Crypto IPv4-ACL Guidelines

Follow these guidelines when configuring IPv4-ACLs for the IPsec feature:

- The Cisco NX-OS software only allows name-based IPv4-ACLs.
- When an IPv4-ACL is applied to a crypto map, the following options apply:
  - Permit—Applies the IPsec feature to the traffic.
  - Deny—Allows clear text (default).

  **Note** IKE traffic (UDP port 500) is implicitly transmitted in clear text.

- The IPsec feature only considers the source and destination IPv4 addresses and subnet masks, protocol, and single port number. There is no support for IPv6 in IPsec.

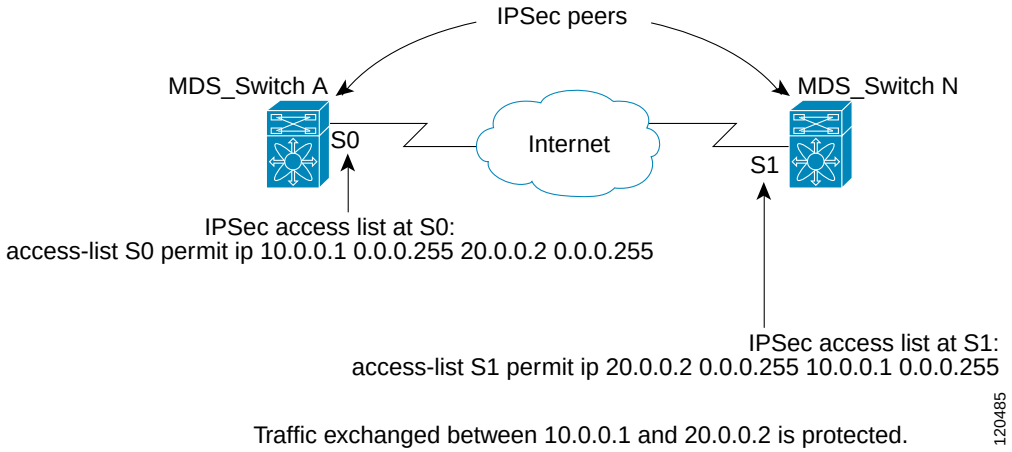
  **Note** The IPsec feature does not support port number ranges and ignores higher port number field, if specified.

- The permit option causes all IP traffic that matches the specified conditions to be protected by crypto, using the policy described by the corresponding crypto map entry.
- The deny option prevents traffic from being protected by crypto. The first deny statement causes the traffic to be in clear text.
- The crypto IPv4-ACL you define is applied to an interface after you define the corresponding crypto map entry and apply the crypto map set to the interface.
- Different IPv4-ACLs must be used in different entries of the same crypto map set.

- Inbound and outbound traffic is evaluated against the same outbound IPv4-ACL. Therefore, the IPv4-ACL's criteria is applied in the forward direction to traffic exiting your switch, and the reverse direction to traffic entering your switch.
- Each IPv4-ACL filter assigned to the crypto map entry is equivalent to one security policy entry.
- IPsec protection (see Figure 7-4) is applied to traffic between switch interface S0 (IPv4 address 10.0.0.1) and switch interface S1 (IPv4 address 20.0.0.2) as the data exits switch A's S0 interface enroute to switch interface S1. For traffic from 10.0.0.1 to 20.0.0.2, the IPv4-ACL entry on switch A is evaluated as follows:
  - source = IPv4 address 10.0.0.1
  - dest = IPv4 address 20.0.0.2

  For traffic from 20.0.0.2 to 10.0.0.1, that same IPv4-ACL entry on switch A is evaluated as follows:
  - source = IPv4 address 20.0.0.2
  - dest = IPv4 address 10.0.0.1

*Figure 7-4 IPsec Processing of Crypto IPv4-ACLs*




- If you configure multiple statements for a given crypto IPv4-ACL that is used for IPsec, the first permit statement that is matched is used to determine the scope of the IPsec SA. Later, if traffic matches a different permit statement of the crypto IPv4-ACL, a new, separate IPsec SA is negotiated to protect traffic matching the newly matched IPv4-ACL statement.
- Unprotected inbound traffic that matches a permit entry in the crypto IPv4-ACL for a crypto map entry flagged as IPsec is dropped, because this traffic was expected to be protected by IPsec.
- You can use the **show ip access-lists** command to view all IP-ACLs. The IP-ACLs used for traffic filtering purposes are also used for crypto.
- For IPsec to interoperate effectively with Microsoft iSCSI initiators, specify the TCP protocol and the local iSCSI TCP port number (default 3260) in the IPv4-ACL. This configuration ensures the speedy recovery of encrypted iSCSI sessions following disruptions such as Gigabit Ethernet interfaces shutdowns, VRRP switchovers, and port failures.
- The following example of a IPv4-ACL entry shows that the MDS switch IPv4 address is 10.10.10.50 and remote Microsoft host running encrypted iSCSI sessions is 10.10.10.16:

```
switch(config)# ip access-list aclmsiscsi2 permit tcp 10.10.10.50 0.0.0.0 range port
3260 3260 10.10.10.16 0.0.0.0
```

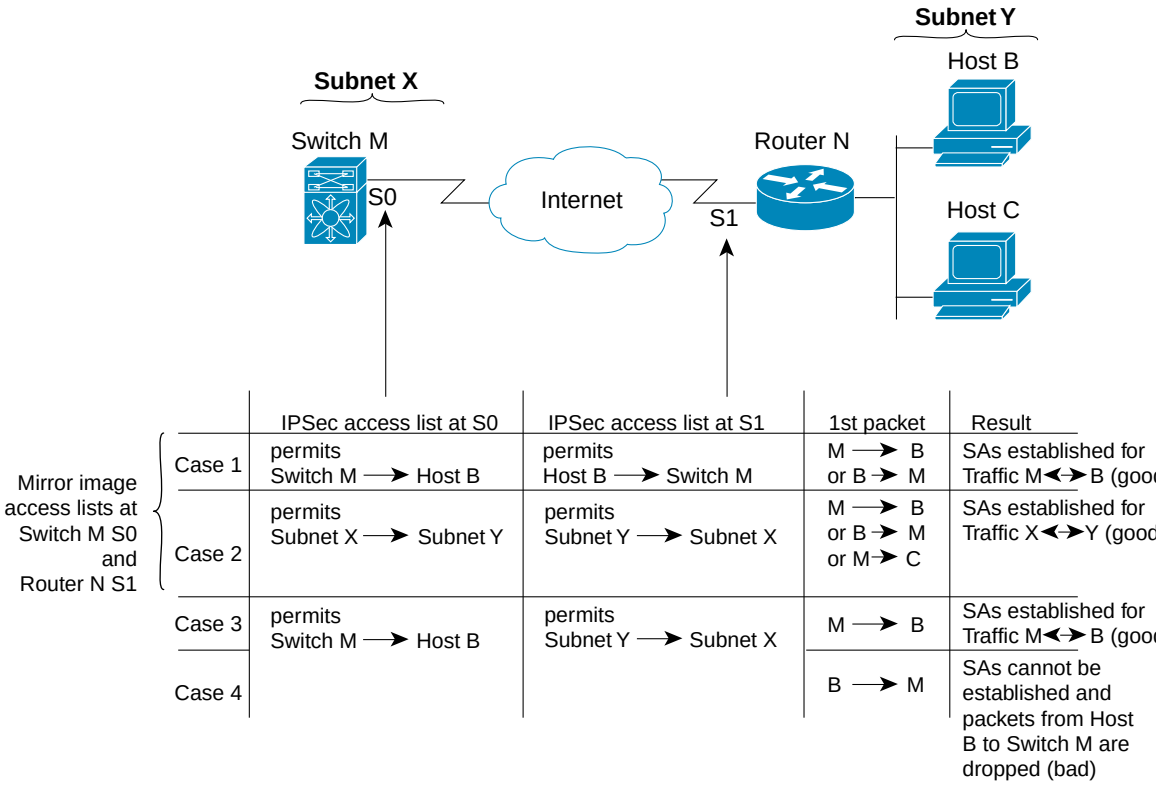
# Mirror Image Crypto IPv4-ACLs

For every crypto IPv4-ACL specified for a crypto map entry defined at the local peer, define a mirror image crypto IPv4-ACL at the remote peer. This configuration ensures that IPsec traffic applied locally can be processed correctly at the remote peer.

**Tip** The crypto map entries themselves must also support common transforms and must refer to the other system as a peer.

Figure 7-5 shows some sample scenarios with and without mirror image IPv4-ACLs.

*Figure 7-5 IPsec Processing of Mirror Image Configuration*




As Figure 7-5 indicates, IPsec SAs can be established as expected whenever the two peers' crypto IPv4-ACLs are mirror images of each other. However, an IPsec SA can be established only some of the time when the IPv4-ACLs are not mirror images of each other. This can happen in the case when an entry in one peer's IPv4-ACL is a subset of an entry in the other peer's IPv4-ACL, such as shown in cases 3 and 4 of Figure 7-5. IPsec SA establishment is critical to IPsec. Without SAs, IPsec does not work, causing any packets matching the crypto IPv4-ACL criteria to be silently dropped instead of being forwarded with IPsec security.

In case 4, an SA cannot be established because SAs are always requested according to the crypto IPv4-ACLs at the initiating packet's end. In case 4, router N requests that all traffic between subnet X and subnet Y be protected, but this is a superset of the specific flows permitted by the crypto IPv4-ACL at switch M so the request is not permitted. Case 3 works because switch M's request is a subset of the specific flows permitted by the crypto IPv4-ACL at router N.

Because of the complexities introduced when crypto IPv4-ACLs are not configured as mirror images at peer IPsec devices, we strongly encourage you to use mirror image crypto IPv4-ACLs.

### The any Keyword in Crypto IPv4-ACLs

**Tip** We recommend that you configure mirror image crypto IPv4-ACLs for use by IPsec and that you avoid using the **any** option.

The **any** keyword in a permit statement is discouraged when you have multicast traffic flowing through the IPsec interface. This configuration can cause multicast traffic to fail.

The **permit any** statement causes all outbound traffic to be protected (and all protected traffic sent to the peer specified in the corresponding crypto map entry) and requires protection for all inbound traffic. Then, all inbound packets that lack IPsec protection are silently dropped, including packets for routing protocols, NTP, echo, echo response, and so forth.

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

## Creating Crypto IPv4-ACLs

To create IPv4-ACLs, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config terminal` | Enters configuration mode. |
| **Step 2** | `switch(config)# ip access-list List1 permit ip 10.1.1.100 0.0.0.255 11.1.1.100 0.0.0.255` | Permits all IP traffic from and to the specified networks. |

**Note** The **show ip access-list** command does not display the crypto map entries. Use the **show crypto map** command to display the associated entries.

Add permit and deny statements as appropriate (see Chapter 5, “Configuring IPv4 and IPv6 Access Control Lists”). Each permit and deny specifies conditions to determine which IP packets must be protected.

## About Transform Sets in IPsec

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

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

During IPsec security association negotiations with IKE, the peers search for a transform set that is the same at both peers. When such a transform set is found, it is selected and applied to the protected traffic as part of both peers' IPsec security associations.

**Tip** If you change a transform set definition, the change is only applied to crypto map entries that reference the transform set. The change is not applied to existing security associations, but used in subsequent negotiations to establish new security associations. If you want the new settings to take effect sooner, you can clear all or part of the security association database.

**Note** When you enable IPsec, the Cisco NX-OS software automatically creates a default transform set (ipsec_default_tranform_set) using AES-128 encryption and SHA-1 authentication algorithms.

Table 7-3 provides a list of allowed transform combinations for IPsec.

*Table 7-3 IPsec Transform Configuration Parameters*

| Parameter | Accepted Values | Keyword |
|---|---|---|
| encryption algorithm | 56-bit DES-CBC | **esp-des** |
| | 168-bit DES | **esp-3des** |
| | 128-bit AES-CBC | **esp-aes 128** |
| | 128-bit AES-CTR[1] | **esp-aes 128 ctr** |
| | 256-bit AES-CBC | **esp-aes 256** |
| | 256-bit AES-CTR[1] | **esp-aes 256 ctr** |
| hash/authentication algorithm[1] (optional) | SHA-1 (HMAC variant) | **esp-sha1-hmac** |
| | SHA-2 (HMAC variant) | **esp-sha256-hmac** |
| | MD5 (HMAC variant) | **esp-sha512-hmac** |
| | AES-XCBC-MAC | **esp-md5-hmac** |
| | | **esp-aes-xcbc-mac**[2] |

1. If you configure the AES counter (CTR) mode, you must also configure the authentication algorithm.
2. Starting from Cisco MDS NX-OS Release 5.2(2), the **esp-aes-xcbc-mac** authentication algorithm is not supported.

The following table lists the supported and verified settings for IPsec and IKE encryption authentication algorithms on the Microsoft Windows and Linux platforms:

| Platform | IKE | IPsec |
|---|---|---|
| Microsoft iSCSI initiator, Microsoft IPsec implementation on Microsoft Windows 2000 platform | 3DES, SHA-1, SHA-2, or MD5, DH group 2 | 3DES, SHA-1, SHA-2 |
| Cisco iSCSI initiator, Free Swan IPsec implementation on Linux platform | 3DES, MD5, DH group 1 | 3DES, MD5 |

# Configuring Transform Sets

To configure transform sets, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config terminal` | Enters configuration mode. |
| **Step 2** | `switch(config)# crypto transform-set domain ipsec test esp-3des esp-md5-hmac` | Configures a transform set called test specifying the 3DES encryption algorithm and the MD5 authentication algorithm. Refer to Table 7-3 to verify the allowed transform combinations. |
| | `switch(config)# crypto transform-set domain ipsec test esp-3des` | Configures a transform set called test specifying the 3DES encryption algorithm. In this case, the default no authentication is performed. |

# About Crypto Map Entries

Once you have created the crypto IPv4-ACLs and transform sets, you can create crypto map entries that combine the various parts of the IPsec SA, including the following:

- The traffic to be protected by IPsec (per the crypto IPv4-ACL). A crypto map set can contain multiple entries, each with a different IPv4-ACL.
- The granularity of the flow to be protected by a set of SAs.
- The IPsec-protected traffic destination (who the remote IPsec peer is).
- The local address to be used for the IPsec traffic (applying to an interface).
- The IPsec security to be applied to this traffic (selecting from a list of one or more transform sets).
- Other parameters to define an IPsec SA.

Crypto map entries with the same crypto map name (but different map sequence numbers) are grouped into a crypto map set.

When you apply a crypto map set to an interface, the following events occur:

- A security policy database (SPD) is created for that interface.
- All IP traffic passing through the interface is evaluated against the SPD.

If a crypto map entry sees outbound IP traffic that requires protection, an SA is negotiated with the remote peer according to the parameters included in the crypto map entry.

The policy derived from the crypto map entries is used during the negotiation of SAs. If the local switch initiates the negotiation, it will use the policy specified in the crypto map entries to create the offer to be sent to the specified IPsec peer. If the IPsec peer initiates the negotiation, the local switch checks the policy from the crypto map entries and decides whether to accept or reject the peer's request (offer).

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

### SA Establishment Between Peers

When two peers try to establish an SA, they must each have at least one crypto map entry that is compatible with one of the other peer's crypto map entries.

For two crypto map entries to be compatible, they must at least meet the following criteria:

- The crypto map entries must contain compatible crypto IPv4-ACLs (for example, mirror image IPv4-ACLs). If the responding peer entry is in the local crypto, the IPv4-ACL must be permitted by the peer's crypto IPv4-ACL.
- The crypto map entries must each identify the other peer or must have auto peer configured.
- If you create more than one crypto map entry for a given interface, use the seq-num of each map entry to rank the map entries: the lower the seq-num, the higher the priority. At the interface that has the crypto map set, traffic is evaluated against higher priority map entries first.
- The crypto map entries must have at least one transform set in common, where IKE negotiations are carried out and SAs are established. During the IPsec SA negotiation, the peers agree to use a particular transform set when protecting a particular data flow.

When a packet matches a permit entry in a particular IPv4-ACL, the corresponding crypto map entry is tagged, and the connections are established.

### Crypto Map Configuration Guidelines

When configuring crypto map entries, follow these guidelines:

- The sequence number for each crypto map decides the order in which the policies are applied. A lower sequence number is assigned a higher priority.
- Only one IPv4-ACL is allowed for each crypto map entry (the IPv4-ACL itself can have multiple permit or deny entries).
- When the tunnel endpoint is the same as the destination address, you can use the auto-peer option to dynamically configure the peer.
- For IPsec to interoperate effectively with Microsoft iSCSI initiators, specify the TCP protocol and the local iSCSI TCP port number (default 3260) in the IPv4-ACL. This configuration ensures the speedy recovery of encrypted iSCSI sessions following disruptions such as Gigabit Ethernet interfaces shutdowns, VRRP switchovers, and port failures.

## Creating Crypto Map Entries

**Note** If the peer IP address specified in the crypto map entry is a VRRP IP address on a remote Cisco MDS switch, ensure that the IP address is created using the **secondary** option (see the *Cisco MDS 9000 Family NX-OS IP Services Configuration Guide* for more information).

To create mandatory crypto map entries, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config terminal` | Enters configuration mode. |
| **Step 2** | `switch(config)# crypto map  domain ipsec SampleMap 31` | Places you in the crypto map configuration mode for the entry named SampleMap with 31 as its sequence number. |
| **Step 3** | `switch(config-crypto-map-ip)# match address SampleAcl` | Names an ACL to determine which traffic should be protected and not protected by IPsec in the context of this crypto map entry. |
| **Step 4** | `switch(config-crypto-map-ip)# set peer 10.1.1.1` | Configures a specific peer IPv4 address.<br>**Note** IKE only supports IPv4 addresses, not IPv6 addresses. |
| **Step 5** | `switch(config-crypto-map-ip)# set transform-set SampleTransform1 SampleTransmfor2` | Specifies which transform sets are allowed for the specified crypto map entry or entries. List multiple transform sets in order of priority (highest priority first). |

# About SA Lifetime Negotiation

You can override the global lifetime values (size and time) by configuring an SA-specific lifetime value.

To specify SA lifetime negotiation values, you can optionally configure the lifetime value for a specified crypto map. If you do, this value overrides the globally set values. If you do not specify the crypto map specific lifetime, the global value (or global default) is used.

See the "Global Lifetime Values" section on page 7-197 for more information on global lifetime values.

# Setting the SA Lifetime

To set the SA lifetime for a specified crypto map entry, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config terminal` | Enters configuration mode. |
| **Step 2** | `switch(config)# crypto map domain ipsec SampleMap 31` | Enters crypto map configuration submode for the entry named SampleMap with 31 as its sequence number. |
| **Step 3** | `switch(config-crypto-map-ip)# set security-association lifetime seconds 8640` | Specifies an SA lifetime for this crypto map entry using different IPsec SA lifetimes than the global lifetimes for the crypto map entry. |
| **Step 4** | `switch(config-crypto-map-ip)# set security-association lifetime gigabytes 4000` | Configures the traffic-volume lifetime for this SA to time out after the specified amount of traffic (in gigabytes) have passed through the FCIP link using the SA. The lifetime ranges from 1 to 4095 gigabytes. |

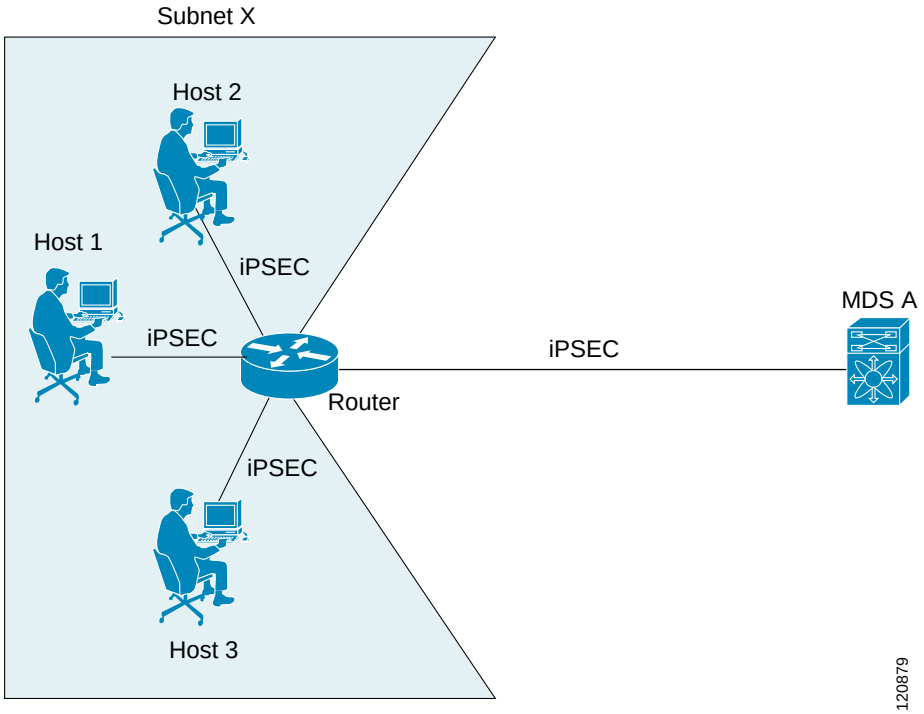
# About the AutoPeer Option

Setting the peer address as **auto-peer** in the crypto map indicates that the destination endpoint of the traffic should be used as the peer address for the SA. Using the same crypto map, a unique SA can be set up at each of the endpoints in the subnet specified by the crypto map's IPv4-ACL entry. Auto-peer simplifies configuration when traffic endpoints are IPsec capable. It is particularly useful for iSCSI, where the iSCSI hosts in the same subnet do not require separate configuration.

Figure 7-6 shows a scenario where the auto-peer option can simplify configuration. Using the auto-peer option, only one crypto map entry is needed for all the hosts from subnet X to set up SAs with the switch. Each host will set up its own SA, but will share the crypto map entry. Without the auto-peer option, each host needs one crypto map entry.

See the “Sample iSCSI Configuration” section on page 7-207 for more details.

*Figure 7-6 iSCSI with End-to-End IPsec Using the auto-peer Option*




# Configuring the AutoPeer Option

To configure the auto-peer option, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config terminal` | Enters configuration mode. |

| | Command | Purpose |
|---|---|---|
| Step 2 | `switch(config)# crypto map domain ipsec SampleMap 31` | Places you in the crypto map configuration mode for the entry named SampleMap with 31 as its sequence number. |
| Step 3 | `switch(config-crypto-map-ip)# set peer auto-peer` | Directs the software to select (during the SA setup) the destination peer IP address dynamically. |

# About Perfect Forward Secrecy

To specify SA lifetime negotiation values, you can also optionally configure the perfect forward secrecy (PFS) value in the crypto map.

The PFS feature is disabled by default. If you set the PFS group, you can set one of the DH groups: 1, 2, 5, or 14. If you do not specify a DH group, the software uses group 1 by default.

# Configuring Perfect Forward Secrecy

To configure the PFS value, follow these steps:

| | Command | Purpose |
|---|---|---|
| Step 1 | `switch# config terminal` | Enters configuration mode. |
| Step 2 | `switch(config)# crypto map domain ipsec SampleMap 31` | Places you in the crypto map configuration mode for the entry named SampleMap with 31 as its sequence number. |
| Step 3 | `switch(config-crypto-map-ip)# set pfs group 2` | Specifies that IPsec should ask for PFS when requesting new SAs for this crypto map entry, or should demand PFS in requests received from the IPsec peer. |

# About Crypto Map Set Interface Application

You need to apply a crypto map set to each interface through which IPsec traffic will flow. Applying the crypto map set to an interface instructs the switch to evaluate all the interface's traffic against the crypto map set and to use the specified policy during connection or SA negotiation on behalf of the traffic to be protected by crypto.

You can apply only one crypto map set to an interface. You can apply the same crypto map to multiple interfaces. However, you cannot apply more than one crypto map set to each interface.

## Applying a Crypto Map Set

To apply a crypto map set to an interface, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config terminal` | Enters configuration mode. |
| **Step 2** | `switch(config)# interface gigabitethernet 4/1` | Selects the required Gigabit Ethernet interface (and subinterface, if required) to which the IPsec crypto map is to be applied. |
| **Step 3** | `switch(config-if)# crypto map domain ipsec cm10` | Applies the crypto map set to the selected interface. |

# IPsec Maintenance

Certain configuration changes will only take effect when negotiating subsequent security associations. If you want the new settings to take immediate effect, you must clear the existing security associations so that they will be reestablished with the changed configuration. If the switch is actively processing IPsec traffic, it is desirable to clear only the portion of the security association database that would be affected by the configuration changes (that is, clear only the security associations established by a given crypto map set). Clearing the full security association database should be reserved for large-scale changes, or when the router is processing very little other IPsec traffic.

**Tip** You can obtain the SA index from the output of the **show crypto sa domain interface gigabitethernet** *slot/port* command.

Use the following command to clear part of the SA database.

```
switch# clear crypto sa domain ipsec interface gigabitethernet 2/1 inbound sa-index 1
```

# Global Lifetime Values

If you have not configured a lifetime in the crypto map entry, the global lifetime values are used when negotiating new IPsec SAs.

You can configure two lifetimes: timed or traffic-volume. An SA expires after the first of these lifetimes is reached. The default lifetimes are 3,600 seconds (one hour) and 450 GB.

If you change a global lifetime, the new lifetime value will not be applied to currently existing SAs, but will be used in the negotiation of subsequently established SAs. If you wish to use the new values immediately, you can clear all or part of the SA database.

Assuming that the particular crypto map entry does not have lifetime values configured, when the switch requests new SAs it will specify its global lifetime values in the request to the peer; it will use this value as the lifetime of the new SAs. When the switch receives a negotiation request from the peer, it uses the value determined by the IKE version in use:

- If you use IKEv1 to set up IPsec SAs, the SA lifetime values are chosen to be the smaller of the two proposals. The same values are programmed on both the ends of the tunnel.

- If you use IKEv2 to set up IPsec SAs, the SAs on each end have their own set up of lifetime values and thus the SAs on both sides expire independently.

The SA (and corresponding keys) will expire according to whichever comes sooner, either after the specified amount of time (in seconds) has passed or after the specified amount of traffic (in bytes) has passed.

A new SA is negotiated before the lifetime threshold of the existing SA is reached to ensure that negotiation completes before the existing SA expires.

The new SA is negotiated when one of the following thresholds is reached (whichever comes first):

- 30 seconds before the lifetime expires or
- Approximately 10% of the lifetime in bytes remain

If no traffic has passed through when the lifetime expires, a new SA is not negotiated. Instead, a new SA will be negotiated only when IPsec sees another packet that should be protected.

To configure global SA lifetimes, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config terminal` | Enters configuration mode. |
| **Step 2** | `switch(config)# crypto global domain ipsec security-association lifetime seconds 86400` | Configures the global timed lifetime for IPsec SAs to time out after the specified number of seconds have passed. The global lifetime ranges from 120 to 86400 seconds. |
| **Step 3** | `switch(config)# crypto global domain ipsec security-association lifetime gigabytes 4000` | Configures the global traffic-volume lifetime for IPsec SAs to time out after the specified amount of traffic (in gigabytes) has passed through the FCIP link using the SA. The global lifetime ranges from 1 to 4095 gigabytes. |
| | `switch(config)# crypto global domain ipsec security-association lifetime kilobytes 2560` | Configures the global traffic-volume lifetime in kilobytes. The global lifetime ranges from 2560 to 2147483647 kilobytes. |
| | `switch(config)# crypto global domain ipsec security-association lifetime megabytes 5000` | Configures the global traffic-volume lifetime in megabytes. The global lifetime ranges from 3 to 4193280 megabytes. |

# Displaying IKE Configurations

You can verify the IKE information by using the **show** set of commands. See Examples 7-1 to 7-5.

***Example 7-1 Displays the Parameters Configured for Each IKE Policy***

```
switch# show crypto ike domain ipsec
keepalive 60000
```

***Example 7-2 Displays the Initiator Configuration***

```
switch# show crypto ike domain ipsec initiator
initiator version 1 address 1.1.1.1
initiator version 1 address 1.1.1.2
```

*Example 7-3 Displays the Key Configuration*

```
switch# show crypto ike domain ipsec key
key abcdefgh address 1.1.1.1
key bcdefghi address 1.1.2.1
```

*Example 7-4 Displays the Currently Established Policies for IKE*

```
switch# show crypto ike domain ipsec policy 1
Priority 1, auth pre-shared, lifetime 6000 secs, encryption 3des, hash md5, DH group 5
Priority 3, auth pre-shared, lifetime 86300 secs, encryption aes, hash sha1, DH group 1
Priority 5, auth pre-shared-key, lifetime 86400 secs, encryption 3des, hash sha256, DH
group 1
```

*Example 7-5 Displays the Currently Established SAs for IKE*

```
switch# show crypto ike domain ipsec sa
Tunn    Local Addr          Remote Addr        Encr     Hash    Auth Method     Lifetime
------------------------------------------------------------------------------------------
1*      172.22.31.165[500]  172.22.31.166[500] 3des     sha1    preshared key   86400
2       172.22.91.174[500]  172.22.91.173[500] 3des     sha1    preshared key   86400
------------------------------------------------------------------------------------------
NOTE: tunnel id ended with * indicates an IKEv1 tunnel
```

# Displaying IPsec Configurations

You can verify the IPsec information by using the **show** set of commands. See Examples 7-6 to 7-19.

*Example 7-6 Displays Information for the Specified ACL*

```
switch# show ip access-list acl10
ip access-list acl10 permit ip 10.10.10.0 0.0.0.255 10.10.10.0 0.0.0.255 (0 matches)
```

In Example 7-6, the display output match is only displayed of an interface (not the crypto map) meets this criteria.

*Example 7-7 Displays the Transform Set Configuration*

```
switch# show crypto transform-set domain ipsec
Transform set: 1/1 {esp-3des esp-sha256-hmac}
    will negotiate {tunnel}
Transform set: ipsec_default_transform_set {esp-aes 128 esp-sha1-hmac}
    will negotiate {tunnel}
```

*Example 7-8 Displays All Configured Crypto Maps*

```
switch# show crypto map domain ipsec
Crypto Map "cm10" 1 ipsec
        Peer = Auto Peer
        IP ACL = acl10
            permit ip 10.10.10.0 255.255.255.0 10.10.10.0 255.255.255.0
        Transform-sets: 3des-md5, des-md5,
        Security Association Lifetime: 4500 megabytes/3600 seconds
        PFS (Y/N): N
        Interface using crypto map set cm10:
```

```
            GigabitEthernet4/1
Crypto Map "cm100" 1 ipsec
        Peer = Auto Peer
        IP ACL = acl100
            permit ip 10.10.100.0 255.255.255.0 10.10.100.0 255.255.255.0
        Transform-sets: 3des-md5, des-md5,
        Security Association Lifetime: 4500 megabytes/3600 seconds
        PFS (Y/N): N
        Interface using crypto map set cm100:
            GigabitEthernet4/2
```

*Example 7-9 Displays the Crypto Map Information for a Specific Interface*

```
switch# show crypto map domain ipsec interface gigabitethernet 4/1
Crypto Map "cm10" 1 ipsec
        Peer = Auto Peer
        IP ACL = acl10
            permit ip 10.10.10.0 255.255.255.0 10.10.10.0 255.255.255.0
        Transform-sets: 3des-md5, des-md5,
        Security Association Lifetime: 4500 megabytes/3600 seconds
        PFS (Y/N): N
        Interface using crypto map set cm10:
            GigabitEthernet4/1
```

*Example 7-10 Displays the Specified Crypto Map Information*

```
switch# show crypto map domain ipsec tag cm100
Crypto Map "cm100" 1 ipsec
        Peer = Auto Peer
        IP ACL = acl100
            permit ip 10.10.100.0 255.255.255.0 10.10.100.0 255.255.255.0
        Transform-sets: 3des-md5, des-md5,
        Security Association Lifetime: 4500 megabytes/3600 seconds
        PFS (Y/N): N
        Interface using crypto map set cm100:
            GigabitEthernet4/2
```

*Example 7-11 Displays SA Association for the Specified Interface*

```
switch# show crypto sad domain ipsec interface gigabitethernet 4/1
interface: GigabitEthernet4/1
    Crypto map tag: cm10, local addr. 10.10.10.1
    protected network:
    local ident (addr/mask): (10.10.10.0/255.255.255.0)
    remote  ident (addr/mask): (10.10.10.4/255.255.255.255)
    current_peer: 10.10.10.4
      local crypto endpt.: 10.10.10.1, remote crypto endpt.: 10.10.10.4
      mode: tunnel, crypto algo: esp-3des, auth algo: esp-md5-hmac
     current outbound spi: 0x30e000f (51249167), index: 0
      lifetimes in seconds:: 3600
      lifetimes in bytes:: 423624704
     current inbound spi: 0x30e0000 (51249152), index: 0
      lifetimes in seconds:: 3600
      lifetimes in bytes:: 423624704
```

*Example 7-12 Displays All SA Associations*

```
switch# show crypto sad domain ipsec
interface: GigabitEthernet4/1
    Crypto map tag: cm10, local addr. 10.10.10.1
    protected network:
    local  ident (addr/mask): (10.10.10.0/255.255.255.0)
    remote  ident (addr/mask): (10.10.10.4/255.255.255.255)
    current_peer: 10.10.10.4
      local crypto endpt.: 10.10.10.1, remote crypto endpt.: 10.10.10.4
      mode: tunnel, crypto algo: esp-3des, auth algo: esp-md5-hmac
     current outbound spi: 0x30e000f (51249167), index: 0
      lifetimes in seconds:: 3600
      lifetimes in bytes:: 423624704
     current inbound spi: 0x30e0000 (51249152), index: 0
      lifetimes in seconds:: 3600
      lifetimes in bytes:: 423624704
```

*Example 7-13 Displays Information About the Policy Database*

```
switch# show crypto spd domain ipsec
Policy Database for interface: GigabitEthernet4/1, direction: Both
#   0:      deny  udp any port eq 500 any
#   1:      deny  udp any any port eq 500
#   2:      permit ip 10.10.10.0 255.255.255.0 10.10.10.0 255.255.255.0
#  63:      deny  ip any any
Policy Database for interface: GigabitEthernet4/2, direction: Both
#   0:      deny  udp any port eq 500 any <----------------------UDP default entry
#   1:      deny  udp any any port eq 500 <---------------------- UDP default entry
#   3:      permit ip 10.10.100.0 255.255.255.0 10.10.100.0 255.255.255.0
#  63:      deny  ip any any <---------------------------------------- Clear text default entry
```

*Example 7-14 Displays SPD Information for a Specific Interface*

```
switch# show crypto spd domain ipsec interface gigabitethernet 4/2
Policy Database for interface: GigabitEthernet3/1, direction: Both
#   0:      deny  udp any port eq 500 any
#   1:      deny  udp any any port eq 500
#   2:      permit ip 10.10.10.0 255.255.255.0 10.10.10.0 255.255.255.0
# 127:      deny  ip any any
```

*Example 7-15 Displays Detailed iSCSI Session Information for a Specific Interface*

```
switch# show iscsi session detail
Initiator iqn.1987-05.com.cisco:01.9f39f09c7468 (ips-host16.cisco.com)
    Initiator ip addr (s): 10.10.10.5
  Session #1 (index 24)
    Discovery session, ISID 00023d000001, Status active

  Session #2 (index 25)
    Target ibm1
    VSAN 1, ISID 00023d000001, TSIH 0, Status active, no reservation
    Type Normal, ExpCmdSN 42, MaxCmdSN 57, Barrier 0
    MaxBurstSize 0, MaxConn 1, DataPDUInOrder Yes
    DataSeqInOrder Yes, InitialR2T Yes, ImmediateData No
    Registered LUN 0, Mapped LUN 0
    Stats:
      PDU: Command: 41, Response: 41
      Bytes: TX: 21388, RX: 0
```

```
    Number of connection: 1
    Connection #1
      iSCSI session is protected by IPSec <----------The iSCSI session protection status
      Local IP address: 10.10.10.4, Peer IP address: 10.10.10.5
      CID 0, State: Full-Feature
      StatSN 43, ExpStatSN 0
      MaxRecvDSLength 131072, our_MaxRecvDSLength 262144
      CSG 3, NSG 3, min_pdu_size 48 (w/ data 48)
      AuthMethod none, HeaderDigest None (len 0), DataDigest None (len 0)
      Version Min: 0, Max: 0
      FC target: Up, Reorder PDU: No, Marker send: No (int 0)
      Received MaxRecvDSLen key: Yes
```

***Example 7-16 Displays FCIP Information for a Specific Interface***

```
switch# show interface fcip 1
fcip1 is trunking
    Hardware is GigabitEthernet
    Port WWN is 20:50:00:0d:ec:08:6c:c0
    Peer port WWN is 20:10:00:05:30:00:a7:9e
    Admin port mode is auto, trunk mode is on
    Port mode is TE
    Port vsan is 1
    Speed is 1 Gbps
    Trunk vsans (admin allowed and active) (1)
    Trunk vsans (up)                       (1)
    Trunk vsans (isolated)                 ()
    Trunk vsans (initializing)             ()
    Using Profile id 1  (interface GigabitEthernet2/1)
    Peer Information
      Peer Internet address is 10.10.11.1 and port is 3225
    FCIP tunnel is protected by IPSec <----------The FCIP tunnel protection status
    Write acceleration mode is off
    Tape acceleration mode is off
    Tape Accelerator flow control buffer size is 256 KBytes
    IP Compression is disabled
    Special Frame is disabled
    Maximum number of TCP connections is 2
    Time Stamp is disabled
    QOS control code point is 0
    QOS data code point is 0
    B-port mode disabled
    TCP Connection Information
      2 Active TCP connections
        Control connection: Local 10.10.11.2:3225, Remote 10.10.11.1:65520
        Data connection: Local 10.10.11.2:3225, Remote 10.10.11.1:65522
      2 Attempts for active connections, 0 close of connections
    TCP Parameters
      Path MTU 1400 bytes
      Current retransmission timeout is 200 ms
      Round trip time: Smoothed 2 ms, Variance: 1
      Advertized window: Current: 124 KB, Maximum: 124 KB, Scale: 6
      Peer receive window: Current: 123 KB, Maximum: 123 KB, Scale: 6
      Congestion window: Current: 53 KB, Slow start threshold: 48 KB
      Current Send Buffer Size: 124 KB, Requested Send Buffer Size: 0 KB
      CWM Burst Size: 50 KB
    5 minutes input rate 128138888 bits/sec, 16017361 bytes/sec, 7937 frames/sec
    5 minutes output rate 179275536 bits/sec, 22409442 bytes/sec, 46481 frames/sec
      10457037 frames input, 21095415496 bytes
         308 Class F frames input, 32920 bytes
         10456729 Class 2/3 frames input, 21095382576 bytes
         9907495 Reass frames
```

```
   0 Error frames timestamp error 0
63792101 frames output, 30250403864 bytes
   472 Class F frames output, 46816 bytes
   63791629 Class 2/3 frames output, 30250357048 bytes
   0 Error frames
```

*Example 7-17 Displays the Global IPsec Statistics for the Switch*

```
switch# show crypto global domain ipsec
IPSec global statistics:
        Number of crypto map sets: 3
        IKE transaction stats: 0 num, 256 max
        Inbound SA stats: 0 num
        Outbound SA stats: 0 num
```

*Example 7-18 Displays the IPsec Statistics for the Specified Interface*

```
switch# show crypto global domain ipsec interface gigabitethernet 3/1
IPSec interface statistics:
        IKE transaction stats: 0 num
        Inbound SA stats: 0 num, 512 max
        Outbound SA stats: 0 num, 512 max
```

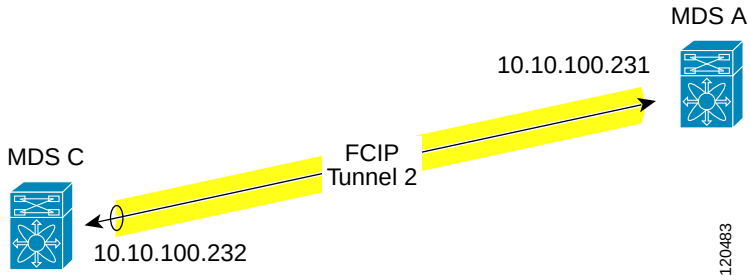
*Example 7-19 Displays the Global SA Lifetime Values*

```
switch# show crypto global domain ipsec security-association lifetime
Security Association Lifetime: 450 gigabytes/3600 seconds
```

# Sample FCIP Configuration

Figure 7-7 focuses on implementing IPsec for one FCIP link (Tunnel 2). Tunnel 2 carries encrypted data between MDS A and MDS C.

*Figure 7-7 IP Security Usage in an FCIP Scenario*




To configure IPsec for the FCIP scenario shown in Figure 7-7, follow these steps:

**Step 1** Enable IKE and IPsec in Switch MDS A.

```
sw10.1.1.100# conf t
sw10.1.1.100(config)# feature crypto ike
sw10.1.1.100(config)# feature crypto ipsec
```

**Step 2** Configure IKE in Switch MDS A.

```
sw10.1.1.100(config)# crypto ike domain ipsec
sw10.1.1.100(config-ike-ipsec)# key ctct address 10.10.100.232
sw10.1.1.100(config-ike-ipsec)# policy 1
sw10.1.1.100(config-ike-ipsec-policy)# encryption 3des
sw10.1.1.100(config-ike-ipsec-policy)# hash md5
sw10.1.1.100(config-ike-ipsec-policy)# end
sw10.1.1.100#
```

**Step 3** Configure the ACLs in Switch MDS A.

```
sw10.1.1.100# conf t
sw10.1.1.100(config)# ip access-list acl1 permit tcp 10.10.100.231 0.0.0.0 10.10.100.232
0.0.0.0
```

**Step 4** Configure the transform set in Switch MDS A.

```
sw10.1.1.100(config)# crypto transform-set domain ipsec tfs-02 esp-aes 128 esp-sha1-hmac
```

**Step 5** Configure the crypto map in Switch MDS A.

```
sw10.1.1.100(config)# crypto map domain ipsec cmap-01 1
sw10.1.1.100(config-crypto-map-ip)# match address acl1
sw10.1.1.100(config-crypto-map-ip)# set peer 10.10.100.232
sw10.1.1.100(config-crypto-map-ip)# set transform-set tfs-02
sw10.1.1.100(config-crypto-map-ip)# set security-association lifetime seconds 3600
sw10.1.1.100(config-crypto-map-ip)# set security-association lifetime gigabytes 3000
sw10.1.1.100(config-crypto-map-ip)# set pfs group5
sw10.1.1.100(config-crypto-map-ip)# end
sw10.1.1.100#
```

**Step 6** Bind the interface to the crypto map set in Switch MDS A.

```
sw10.1.1.100# conf t
sw10.1.1.100(config)# int gigabitethernet 7/1
sw10.1.1.100(config-if)# ip addr 10.10.100.231 255.255.255.0
sw10.1.1.100(config-if)# crypto map domain ipsec cmap-01
sw10.1.1.100(config-if)# no shut
sw10.1.1.100(config-if)# exit
sw10.1.1.100(config)#
```

**Step 7** Configure FCIP in Switch MDS A.

```
sw10.1.1.100(config)# feature fcip
sw10.1.1.100(config)# fcip profile 2
sw10.1.1.100(config-profile)# ip address 10.10.100.231
sw10.1.1.100(config-profile)# int fcip 2
sw10.1.1.100(config-if)# peer-info ipaddr 10.10.100.232
sw10.1.1.100(config-if)# use-profile 2
sw10.1.1.100(config-if)# no shut
sw10.1.1.100(config-if)# end
sw10.1.1.100#
```

**Step 8** Verify the configuration in Switch MDS A.

```
sw10.1.1.100# show crypto global domain ipsec security-association lifetime
Security Association Lifetime: 4500 megabytes/3600 seconds

sw10.1.1.100# show crypto map domain ipsec
Crypto Map "cmap-01" 1 ipsec
        Peer = 10.10.100.232
        IP ACL = acl1
            permit ip 10.10.100.231 255.255.255.255 10.10.100.232 255.255.255.255
        Transform-sets: tfs-02,
        Security Association Lifetime: 3000 gigabytes/3600 seconds
        PFS (Y/N): Y
```

```
        PFS Group: group5
Interface using crypto map set cmap-01:
    GigabitEthernet7/1

sw10.1.1.100# show crypto transform-set domain ipsec
Transform set: tfs-02 {esp-aes 128 esp-sha1-hmac}
    will negotiate {tunnel}

sw10.1.1.100# show crypto spd domain ipsec
Policy Database for interface: GigabitEthernet7/1, direction: Both
#   0:      deny  udp any port eq 500 any
#   1:      deny  udp any any port eq 500
#   2:      permit ip 10.10.100.231 255.255.255.255 10.10.100.232 255.255.255.255
#  63:      deny  ip any any

sw10.1.1.100# show crypto ike domain ipsec
keepalive 3600

sw10.1.1.100# show crypto ike domain ipsec key
key ctct address 10.10.100.232

sw10.1.1.100# show crypto ike domain ipsec policy
Priority 1, auth pre-shared, lifetime 86300 secs, encryption 3des, hash md5, DH group 1
```

**Step 9** Enable IKE and IPsec in Switch MDS C.

```
sw11.1.1.100# conf t
sw11.1.1.100(config)# feature crypto ike
sw11.1.1.100(config)# feature crypto ipsec
```

**Step 10** Configure IKE in Switch MDS C.

```
sw11.1.1.100(config)# crypto ike domain ipsec
sw11.1.1.100(config-ike-ipsec)# key ctct address 10.10.100.231
sw11.1.1.100(config-ike-ipsec)# policy 1
sw11.1.1.100(config-ike-ipsec-policy)# encryption 3des
sw11.1.1.100(config-ike-ipsec-policy)# hash md5
sw11.1.1.100(config-ike-ipsec-policy)# exit
sw11.1.1.100(config-ike-ipsec)# end
sw11.1.1.100#
```

**Step 11** Configure the ACLs in Switch MDS C.

```
sw11.1.1.100# conf t
sw11.1.1.100(config)# ip access-list acl1 permit ip 10.10.100.232 0.0.0.0 10.10.100.231
0.0.0.0
```

**Step 12** Configure the transform set in Switch MDS C.

```
sw11.1.1.100(config)# crypto transform-set domain ipsec tfs-02 esp-aes 128 esp-sha1-hmac
```

**Step 13** Configure the crypto map in Switch MDS C.

```
sw11.1.1.100(config)# crypto map domain ipsec cmap-01 1
sw11.1.1.100(config-crypto-map-ip)# match address acl1
sw11.1.1.100(config-crypto-map-ip)# set peer 10.10.100.231
sw11.1.1.100(config-crypto-map-ip)# set transform-set tfs-02
sw11.1.1.100(config-crypto-map-ip)# set security-association lifetime seconds 3600
sw11.1.1.100(config-crypto-map-ip)# set security-association lifetime gigabytes 3000
sw11.1.1.100(config-crypto-map-ip)# set pfs group5
sw11.1.1.100(config-crypto-map-ip)# exit
sw11.1.1.100(config)#
```

**Step 14** Bind the interface to the crypto map set in Switch MDS C.

```
sw11.1.1.100(config)# int gigabitethernet 1/2
sw11.1.1.100(config-if)# ip addr 10.10.100.232 255.255.255.0
sw11.1.1.100(config-if)# crypto map domain ipsec cmap-01
sw11.1.1.100(config-if)# no shut
sw11.1.1.100(config-if)# exit
sw11.1.1.100(config)#
```

**Step 15** Configure FCIP in Switch MDS C.

```
sw11.1.1.100(config)# feature fcip
sw11.1.1.100(config)# fcip profile 2
sw11.1.1.100(config-profile)# ip address 10.10.100.232
sw11.1.1.100(config-profile)# int fcip 2
sw11.1.1.100(config-if)# peer-info ipaddr 10.10.100.231
sw11.1.1.100(config-if)# use-profile 2
sw11.1.1.100(config-if)# no shut
sw11.1.1.100(config-if)# exit
sw11.1.1.100(config)# exit
```

**Step 16** Verify the configuration in Switch MDS C.

```
sw11.1.1.100# show crypto global domain ipsec security-association lifetime
Security Association Lifetime: 4500 megabytes/3600 seconds

sw11.1.1.100# show crypto map domain ipsec
Crypto Map "cmap-01" 1 ipsec
        Peer = 10.10.100.231
        IP ACL = acl1
            permit ip 10.10.100.232 255.255.255.255 10.10.100.231 255.255.255.255
        Transform-sets: tfs-02,
        Security Association Lifetime: 3000 gigabytes/3600 seconds
        PFS (Y/N): Y
         PFS Group: group5
Interface using crypto map set cmap-01:
    GigabitEthernet1/2

sw11.1.1.100# show crypto spd domain ipsec
Policy Database for interface: GigabitEthernet1/2, direction: Both
#   0:      deny  udp any port eq 500 any
#   1:      deny  udp any any port eq 500
#   2:      permit ip 10.10.100.232 255.255.255.255 10.10.100.231 255.255.255.255
#  63:      deny  ip any any

sw11.1.1.100# show crypto sad domain ipsec
interface: GigabitEthernet1/2
    Crypto map tag: cmap-01, local addr. 10.10.100.232
    protected network:
    local  ident (addr/mask): (10.10.100.232/255.255.255.255)
    remote  ident (addr/mask): (10.10.100.231/255.255.255.255)
    current_peer: 10.10.100.231
      local crypto endpt.: 10.10.100.232, remote crypto endpt.: 10.10.100.231
      mode: tunnel, crypto algo: esp-3des, auth algo: esp-md5-hmac
     current outbound spi: 0x38f96001 (955867137), index: 29
      lifetimes in seconds:: 3600
      lifetimes in bytes:: 3221225472000
     current inbound spi: 0x900b011 (151040017), index: 16
      lifetimes in seconds:: 3600
      lifetimes in bytes:: 3221225472000

sw11.1.1.100# show crypto transform-set domain ipsec
Transform set: tfs-02 {esp-aes 128 esp-sha1-hmac}
    will negotiate {tunnel}

sw11.1.1.100# show crypto ike domain ipsec
```

```
keepalive 3600

sw11.1.1.100# show crypto ike domain ipsec key

key ctct address 10.10.100.231

sw11.1.1.100# show crypto ike domain ipsec policy
Priority 1, auth pre-shared, lifetime 86300 secs, encryption 3des, hash md5, DH
group 1

sw11.1.1.100# show crypto ike domain ipsec sa
Tunn    Local Addr          Remote Addr         Encr    Hash    Auth Method     Lifetime
------------------------------------------------------------------------------------------
1*      10.10.100.232[500]  10.10.100.231[500]  3des    md5     preshared key   86300
------------------------------------------------------------------------------------------
NOTE: tunnel id ended with * indicates an IKEv1 tunnel
```

**Step 17** Verify the configuration in Switch MDS A.

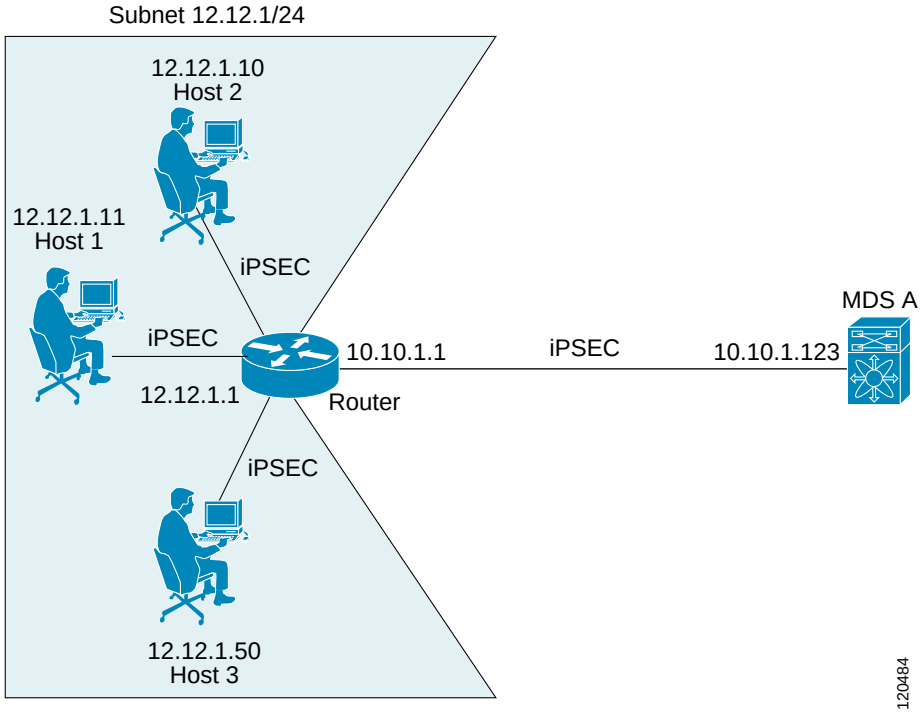
```
sw10.1.1.100# show crypto sad domain ipsec
interface: GigabitEthernet7/1
    Crypto map tag: cmap-01, local addr. 10.10.100.231
    protected network:
    local  ident (addr/mask): (10.10.100.231/255.255.255.255)
    remote  ident (addr/mask): (10.10.100.232/255.255.255.255)
    current_peer: 10.10.100.232
      local crypto endpt.: 10.10.100.231, remote crypto endpt.: 10.10.100.232
      mode: tunnel, crypto algo: esp-3des, auth algo: esp-md5-hmac
     current outbound spi: 0x900b01e (151040030), index: 10
      lifetimes in seconds:: 3600
      lifetimes in bytes:: 3221225472000
     current inbound spi: 0x38fe700e (956198926), index: 13
      lifetimes in seconds:: 3600
      lifetimes in bytes:: 3221225472000

sw10.1.1.100# show crypto ike domain ipsec sa
Tunn Local Addr          Remote Addr         Encr  Hash  Auth Method   Lifetime
------------------------------------------------------------------------------
   1 10.10.100.231[500]  10.10.100.232[500]  3des  md5   preshared key    86300
```

You have now configured IPsec in both switches MDS A and MDS C.

# Sample iSCSI Configuration

Figure 7-8 focuses on the iSCSI session between MDS A and the hosts in subnet 12.12.1/24. Using the **auto-peer** option, when any host from the subnet 12.12.1.0/24 tries to connect to the MDS switch's Gigabit Ethernet port 7/1, an SA is created between the hosts and the MDS switch. With auto-peer, only one crypto map is necessary to create SAs for all the hosts in the same subnet. Without auto-peer, you need one crypto map entry per host.

*Figure 7-8 iSCSI with End-to-End IPsec*




To configure IPsec for the iSCSI scenario shown in Figure 7-8, follow these steps:

**Step 1** Configure the ACLs in Switch MDS A.

```
sw10.1.1.100# conf t
sw10.1.1.100(config)# ip access-list acl1 permit tcp 10.10.1.0 0.0.0.255 range port 3260
3260 12.12.1.0 0.0.0.255
```

**Step 2** Configure the transform set in Switch MDS A.

```
sw10.1.1.100(config)# crypto transform-set domain ipsec tfs-01 esp-3des esp-md5-hmac
```

**Step 3** Configure the crypto map in Switch MDS A.

```
sw10.1.1.100(config)# crypto map domain ipsec cmap-01 1
sw10.1.1.100(config-crypto-map-ip)# match address acl1
sw10.1.1.100(config-crypto-map-ip)# set peer auto-peer
sw10.1.1.100(config-crypto-map-ip)# set transform-set tfs-01
sw10.1.1.100(config-crypto-map-ip)# end
sw10.1.1.100#
```

**Step 4** Bind the interface to the crypto map set in Switch MDS A.

```
sw10.1.1.100# conf t
sw10.1.1.100(config)# int gigabitethernet 7/1
sw10.1.1.100(config-if)# ip address 10.10.1.123 255.255.255.0
sw10.1.1.100(config-if)# crypto map domain ipsec cmap-01
sw10.1.1.100(config-if)# no shut
sw10.1.1.100(config-if)# end
sw10.1.1.100#
```

You have now configured IPsec in MDS A using the Cisco MDS IPsec and iSCSI features.

# Default Settings

Table 7-4 lists the default settings for IKE parameters.

*Table 7-4 Default IKE Parameters*

| Parameters | Default |
|---|---|
| IKE | Disabled. |
| IKE version | IKE version 2. |
| IKE encryption algorithm | 3DES. |
| IKE hash algorithm | SHA. |
| IKE authentication method | Preshared keys. |
| IKE DH group identifier | Group 1. |
| IKE lifetime association | 86,400 seconds (equals 24 hours). |
| IKE keepalive time for each peer (v2) | 3,600 seconds (equals 1 hour). |

Table 7-5 lists the default settings for IPsec parameters.

*Table 7-5 Default IPsec Parameters*

| Parameters | Default |
|---|---|
| IPsec | Disabled. |
| Applying IPsec to the traffic. | Deny—allowing clear text. |
| IPsec PFS | Disabled. |
| IPsec global lifetime (traffic-volume) | 450 Gigabytes. |
| IPsec global lifetime (time) | 3,600 seconds (one hour). |


# Configuring FC-SP and DHCHAP

This chapter includes the following sections:



## About Fabric Authentication

Fibre Channel Security Protocol (FC-SP) capabilities provide switch-switch and host-switch authentication to overcome security challenges for enterprise-wide fabrics. Diffie-Hellman Challenge Handshake Authentication Protocol (DHCHAP) is an FC-SP protocol that provides authentication between Cisco MDS 9000 Family switches and other devices. DHCHAP consists of the CHAP protocol combined with the Diffie-Hellman exchange.

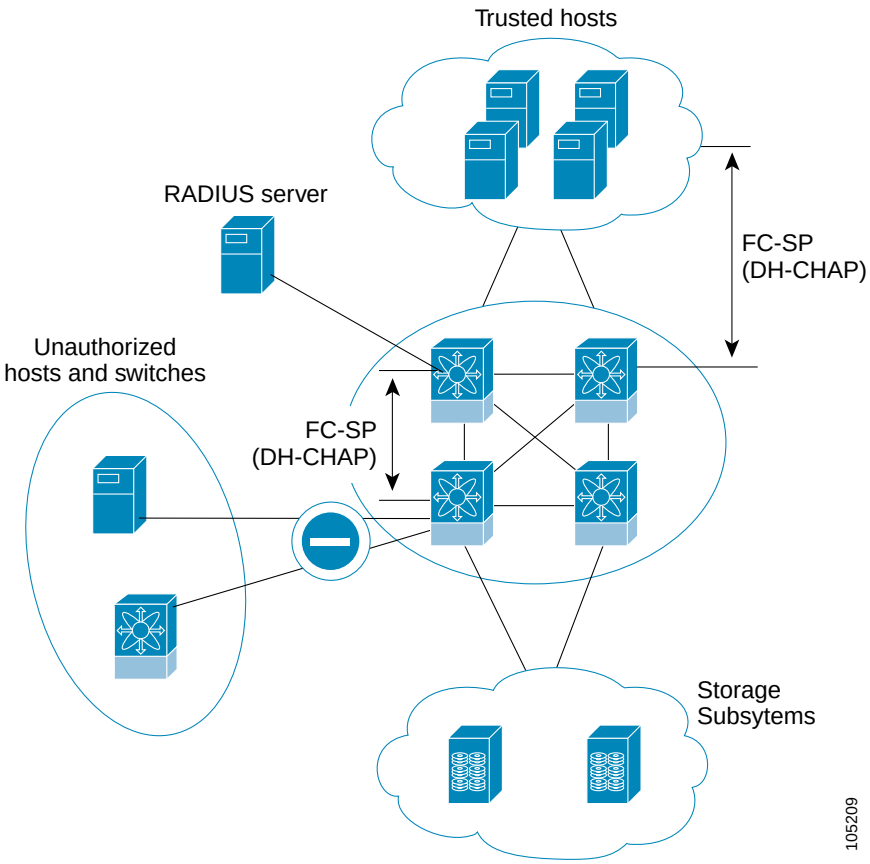
**Note** Cisco NX-OS Release 6.2(1) does not support the Fibre Channel Security Protocol (FC-SP) feature only on Cisco MDS 9710. Support for FC-SP on Cisco MDS 9710 begins in Cisco NX-OS Release 6.2(9).

To authenticate through VFC ports, FC-SP peers use the port VSAN for communication. Hence, the port VSAN needs to be the same and active on both the peers to send and receive authentication messages.

All switches in the Cisco MDS 9000 Family enable fabric-wide authentication from one switch to another switch, or from a switch to a host. These switch and host authentications are performed locally or remotely in each fabric. As storage islands are consolidated and migrated to enterprise-wide fabrics new security challenges arise. The approach of securing storage islands cannot always be guaranteed in enterprise-wide fabrics.

For example, in a campus environment with geographically distributed switches someone could maliciously interconnect incompatible switches or you could accidentally do so, resulting in Inter-Switch Link (ISL) isolation and link disruption. This need for physical security is addressed by switches in the Cisco MDS 9000 Family (see Figure 8-1).

*Figure 8-1 Switch and Host Authentication*




**Note** Fibre Channel (FC) host bus adapters (HBAs) with appropriate firmware and drivers are required for host-switch authentication.

# DHCHAP

DHCHAP is an authentication protocol that authenticates the devices connecting to a switch. Fibre Channel authentication allows only trusted devices to be added to a fabric, which prevents unauthorized devices from accessing the switch.

**Note** The terms FC-SP and DHCHAP are used interchangeably in this chapter.

DHCHAP is a mandatory password-based, key-exchange authentication protocol that supports both switch-to-switch and host-to-switch authentication. DHCHAP negotiates hash algorithms and DH groups before performing authentication. It supports MD5 and SHA-1 algorithm-based authentication.

Configuring the DHCHAP feature requires the ENTERPRISE_PKG license (see the *Cisco MDS 9000 Family NX-OS Licensing Guide*).

To configure DHCHAP authentication using the local password database, follow these steps:

**Step 1** Enable DHCHAP.

**Step 2** Identify and configure the DHCHAP authentication modes.

**Step 3** Configure the hash algorithm and DH group.

**Step 4** Configure the DHCHAP password for the local switch and other switches in the fabric.

**Step 5** Configure the DHCHAP timeout value for reauthentication.

**Step 6** Verify the DHCHAP configuration.

This section includes the following topics:



# DHCHAP Compatibility with Existing Cisco MDS Features

This section identifies the impact of configuring the DHCHAP feature along with existing Cisco MDS features:

- PortChannel interfaces—If DHCHAP is enabled for ports belonging to a PortChannel, DHCHAP authentication is performed at the physical interface level, not at the PortChannel level.
- FCIP interfaces—The DHCHAP protocol works with the FCIP interface just as it would with a physical interface.
- Port security or fabric binding—Fabric binding policies are enforced based on identities authenticated by DHCHAP.
- VSANs—DHCHAP authentication is not done on a per-VSAN basis.
- High availability—DHCHAP authentication works transparently with existing HA features.

# About Enabling DHCHAP

By default, the DHCHAP feature is disabled in all switches in the Cisco MDS 9000 Family.

You must explicitly enable the DHCHAP feature to access the configuration and verification commands for fabric authentication. When you disable this feature, all related configurations are automatically discarded.

# Enabling DHCHAP

To enable DHCHAP for a Cisco MDS switch, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# feature fcsp` | Enables the DHCHAP in this switch. |
| | `switch(config)# no feature fcsp` | Disables (default) the DHCHAP in this switch. |

# About DHCHAP Authentication Modes

The DHCHAP authentication status for each interface depends on the configured DHCHAP port mode.

When the DHCHAP feature is enabled in a switch, each Fibre Channel interface or FCIP interface may be configured to be in one of four DHCHAP port modes:

- On—During switch initialization, if the connecting device supports DHCHAP authentication, the software performs the authentication sequence. If the connecting device does not support DHCHAP authentication, the software moves the link to an isolated state.
- Auto-Active—During switch initialization, if the connecting device supports DHCHAP authentication, the software performs the authentication sequence. If the connecting device does not support DHCHAP authentication, the software continues with the rest of the initialization sequence.
- Auto-Passive (default)—The switch does not initiate DHCHAP authentication, but participates in DHCHAP authentication if the connecting device initiates DHCHAP authentication.
- Off—The switch does not support DHCHAP authentication. Authentication messages sent to such ports return error messages to the initiating switch.

**Note** Whenever DHCHAP port mode is changed to a mode other than the Off mode, reauthentication is performed.

**Note** Changing DHCHAP port mode for a VE link requires a port flap on both the ends.

Table 8-1 identifies the switch-to-switch authentication behavior between two Cisco MDS switches in various modes.

*Table 8-1 DHCHAP Authentication Status Between Two MDS Switches*

| Switch N DHCHAP Modes | Switch 1 DHCHAP Modes | | | |
|---|---|---|---|---|
| | **on** | **auto-active** | **auto-passive** | **off** |
| **on** | FC-SP authentication is performed. | FC-SP authentication is performed. | FC-SP authentication is performed. | Link is brought down. |
| **auto-Active** | | | | FC-SP authentication is *not* performed. |
| **auto-Passive** | | | FC-SP authentication is *not* performed. | |
| **off** | Link is brought down. | FC-SP authentication is *not* performed. | | |

# Configuring the DHCHAP Mode

To configure the DHCHAP mode for a particular interface, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# configure terminal` | Enters configuration mode. |
| **Step 2** | `switch(config)# interface fc2/1-3`<br>`switch(config-if)#` | Selects a range of interfaces and enters the interface configuration submode. |
| **Step 3** | `switch(config-if)# fcsp on` | Sets the DHCHAP mode for the selected interfaces to be in the on state. |
| | `switch(config-if)# no fcsp on` | Reverts to the factory default of auto-passive for these three interfaces. |
| **Step 4** | `switch(config-if)# fcsp auto-active 0` | Changes the DHCHAP authentication mode for the selected interfaces to auto-active. Zero (0) indicates that the port does not perform reauthentication. |
| | `switch(config-if)# fcsp auto-active 120` | Changes the DHCHAP authentication mode to auto-active for the selected interfaces and enables reauthentication every two hours (120 minutes) after the initial authentication. |
| | `switch(config-if)# fcsp auto-active` | Changes the DHCHAP authentication mode to auto-active for the selected interfaces. Reauthentication is disabled (default). |

# About DHCHAP Hash Algorithm

Cisco MDS switches support a default hash algorithm priority list of MD5 followed by SHA-1 for DHCHAP authentication.

**Tip** If you change the hash algorithm configuration, then change it globally for all switches in the fabric.

**Caution** If AAA authentication for fcsp dhchap is enabled, the MD5 hash algorithm must be set if AAA authentication uses RADIUS or TACACS+. This is because RADIUS and TACACS+ applications do not support other hash algorithms.

# Configuring the DHCHAP Hash Algorithm

To configure the hash algorithm, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# configure terminal` | Enters configuration mode. |
| **Step 2** | `switch(config)# fcsp dhchap hash sha1` | Configures the use of only the SHA-1 hash algorithm. |
| | `switch(config)# fcsp dhchap hash MD5` | Configures the use of only the MD5 hash algorithm. |
| | `switch(config)# fcsp dhchap hash md5 sha1` | Defines the use of the default hash algorithm priority list of MD5 followed by SHA-1 for DHCHAP authentication. |
| | `switch(config)# no fcsp dhchap hash sha1` | Reverts to the factory default priority list of the MD5 hash algorithm followed by the SHA-1 hash algorithm. |

# About DHCHAP Group Settings

FC-SP supports multiple DHCHAP groups. The allowed groups may be changed from the default list. The list is configured in the order of highest to lowest priority to be used when negotiating with the FC-SP peer. Each side compares the list of groups received with the local group list and the highest priority group is used. Each group should be specified no more than once in the configuration command.

Refer to the fcsp dhchap command in the *Cisco MDS 9000 Series NX-OS Command Reference Guide* for details about the groups.

**Tip** If you change the DH group configuration, change it globally for all switches in the fabric.

# Configuring the DHCHAP Group Settings

To change the DH group settings, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# configure terminal` | Enters configuration mode. |
| **Step 2** | `switch(config)# fcsp dhchap dhgroup 2 3 4` | Specifies the list of DH groups to be use. The list is specified in order of descending priority. Unspecified groups are excluded from use by DHCHAP. |
| | `switch(config)# no fcsp dhchap dhgroup 2 3 4` | Reverts to the DHCHAP default order. |

# About DHCHAP Password

DHCHAP authentication in each direction requires a shared secret password between the connected devices. To do this, you can use one of three approaches to manage passwords for all switches in the fabric that participate in DHCHAP.

- Approach 1—Use the same password for all switches in the fabric. This is the simplest approach. When you add a new switch, you use the same password to authenticate that switch in this fabric. It is also the most vulnerable approach if someone from the outside maliciously attempts to access any one switch in the fabric.
- Approach 2—Use a different password for each switch and maintain that password list in each switch in the fabric. When you add a new switch, you create a new password list and update all switches with the new list. Accessing one switch yields the password list for all switches in that fabric.
- Approach 3—Use different passwords for different switches in the fabric. When you add a new switch, multiple new passwords corresponding to each switch in the fabric must be generated and configured in each switch. Even if one switch is compromised, the password of other switches are still protected. This approach requires considerable password maintenance by the user.

**Note** All passwords are restricted to 64 alphanumeric characters and can be changed, but not deleted.

**Tip** We recommend using RADIUS or TACACS+ for fabrics with more than five switches. If you need to use a local password database, you can continue to do so using Approach 3 and using the Cisco MDS 9000 Family Fabric Manager to manage the password database.

# Configuring DHCHAP Passwords for the Local Switch

To configure the DHCHAP password for the local switch, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |

| | Command | Purpose |
|---|---|---|
| Step 2 | `switch(config)# fcsp dhchap password 0 mypassword` | Configures a clear text password for the local switch. |
| | `switch(config)# fcsp dhchap password 0 mypassword 30:11:bb:cc:dd:33:11:22` | Configures a clear text password for the local switch to be used for the device with the specified WWN. |
| | `switch(config)# no fcsp dhchap password 0 mypassword 30:11:bb:cc:dd:33:11:22` | Removes the clear text password for the local switch to be used for the device with the specified WWN. |
| | `switch(config)# fcsp dhchap password 7 sfsfdf` | Configures a password entered in an encrypted format for the local switch. |
| | `switch(config)# fcsp dhchap password 7 sfsfdf 29:11:bb:cc:dd:33:11:22` | Configures a password entered in an encrypted format for the local switch to be used for the device with the specified WWN. |
| | `switch(config)# no fcsp dhchap password 7 sfsfdf 29:11:bb:cc:dd:33:11:22` | Removes the password entered in an encrypted format for the local switch to be used for the device with the specified WWN. |
| | `switch(config)# fcsp dhchap password mypassword1` | Configures a clear text password for the local switch to be used with any connecting device. |

To configure the DHCHAP password for the local switch using Fabric Manager, follow these steps:

**Step 1** Expand **Switches > Security** and then select **FC-SP**.

You see the FC-SP configuration in the Information pane.

**Step 2** Click the **Local Passwords** tab.

**Step 3** Click the **Create Row** icon to create a new local password.

You see the Create Local Passwords dialog box.

**Step 4** (Optional) Check the switches that you want to configure the same local password on.

**Step 5** Select the switch WNN and fill in the Password field.

**Step 6** Click **Create** to save the updated password.

# About Password Configuration for Remote Devices

You can configure passwords in the local authentication database for other devices in a fabric. The other devices are identified by their device name, which is also known as the switch WWN or device WWN. The password is restricted to 64 characters and can be specified in clear text (0) or in encrypted text (7).

**Note** The switch WWN identifies the physical switch. This WWN is used to authenticate the switch and is different from the VSAN node WWN.

# Configuring DHCHAP Passwords for Remote Devices

To locally configure the remote DHCHAP password for another switch in the fabric, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# fcsp dhchap devicename 00:11:22:33:44:aa:bb:cc password NewPassword` | Configures a password for another switch in the fabric that is identified by the switch WWN device name. |
| | `switch(config)# no fcsp dhchap devicename 00:11:22:33:44:aa:bb:cc password NewPassword` | Removes the password entry for this switch from the local authentication database. |
| | `switch(config)# fcsp dhchap devicename 00:11:55:66:00:aa:bb:cc password 0 NewPassword` | Configures a clear text password for another switch in the fabric that is identified by the switch WWN device name. |
| | `switch(config)# fcsp dhchap devicename 00:11:22:33:55:aa:bb:cc password 7 asdflkjh` | Configures a password entered in an encrypted format for another switch in the fabric that is identified by the switch WWN device name. |

# About DHCHAP Timeout Value

During the DHCHAP protocol exchange, if the MDS switch does not receive the expected DHCHAP message within a specified time interval, authentication failure is assumed. The time ranges from 20 (no authentication is performed) to 1000 seconds. The default is 30 seconds.

When changing the timeout value, consider the following factors:

- The existing RADIUS and TACACS+ timeout values.
- The same value must also be configured on all switches in the fabric.

# Configuring the DHCHAP Timeout Value

To configure the DHCHAP timeout value, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# fcsp timeout 60` | Configures the reauthentication timeout to be 60 seconds. |
| | `switch(config)# no fcsp timeout 60` | Reverts to the factory default of 30 seconds. |

# Configuring DHCHAP AAA Authentication

You can individually set authentication options. If authentication is not configured, local authentication is used by default.

To configure the AAA authentication refer to the Chapter 4, "Configuring Security Features on an External AAA Server." Follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# aaa authentication dhchap default group TacacsServer1` | Enables DHCHAP to use the TACACS+ server group (in this example, TacacsServer1) for authentication. |
| | `switch(config)# aaa authentication dhchap default local` | Enables DHCHAP for local authentication. |
| | `switch(config)# aaa authentication dhchap default group RadiusServer1` | Enables DHCHAP to use the RADIUS server group (in this example, RadiusServer1) for authentication. |

# Displaying Protocol Security Information

Use the **show fcsp** commands to display configurations for the local database (see Example 8-1 through 8-6).

***Example 8-1 Displays DHCHAP Configurations in FC Interfaces***

```
switch# show fcsp interface fc1/9

fc1/9:
        fcsp authentication mode:SEC_MODE_ON
        Status: Successfully authenticated
```

***Example 8-2 Displays DHCHAP Statistics for an FC Interface***

```
switch# show fcsp interface fc1/9 statistics

fc1/9:
        fcsp authentication mode:SEC_MODE_ON
        Status: Successfully authenticated
        Statistics:
        FC-SP Authentication Succeeded:5
        FC-SP Authentication Failed:0
        FC-SP Authentication Bypassed:0
```

***Example 8-3 Displays the FC-SP WWN of the Device Connected through a Specified Interface***

```
switch# show fcsp interface fc 2/1 wwn

fc2/1:
        fcsp authentication mode:SEC_MODE_ON
        Status: Successfully authenticated
        Other device's WWN:20:00:00:e0:8b:0a:5d:e7
```

***Example 8-4 Displays Hash Algorithm and DHCHAP Groups Configured for the Local Switch***

```
switch# show fcsp dhchap
Supported Hash algorithms (in order of preference):
DHCHAP_HASH_MD5
DHCHAP_HASH_SHA_1
```

```
Supported Diffie Hellman group ids (in order of preference):
DHCHAP_GROUP_NULL
DHCHAP_GROUP_1536
DHCHAP_GROUP_1024
DHCHAP_GROUP_1280
DHCHAP_GROUP_2048
```

***Example 8-5 Displays the DHCHAP Local Password Database***

```
switch# show fcsp dhchap database
DHCHAP Local Password:
        Non-device specific password:**********
        Password for device with WWN:29:11:bb:cc:dd:33:11:22 is ******
        Password for device with WWN:30:11:bb:cc:dd:33:11:22 is *********

Other Devices' Passwords:
        Password for device with WWN:00:11:22:33:44:aa:bb:cc is ***********
```

***Example 8-6 Displays the ASCII Representation of the Device WWN***

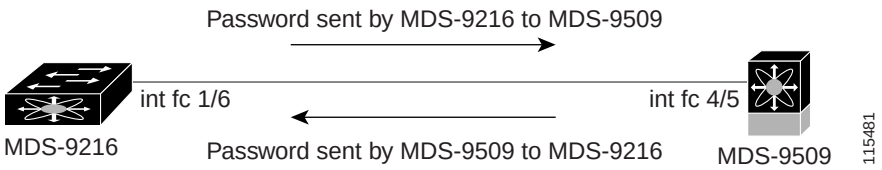
```
switch# show fcsp asciiwwn 30:11:bb:cc:dd:33:11:22
Ascii representation of WWN to be used with AAA servers:Ox_3011bbccdd331122
```

**Tip** Use the ASCII representation of the device WWN (identified in bold in Example 8-6) to configure the switch information on RADIUS and TACACS+ servers.

# Sample Configuration

This section provides the steps to configure the example illustrated in Figure 8-2.

***Figure 8-2 Sample DHCHAP Authentication***




To configure the authentication setup shown in Figure 8-2, follow these steps:

**Step 1** Obtain the device name of the MDS 9216 Switch in the fabric. The MDS 9216 Switch in the fabric is identified by the switch WWN.

```
MDS-9216# show wwn switch
Switch WWN is 20:00:00:05:30:00:54:de
```

**Step 2** Explicitly enable DHCHAP in this switch.

**Note** When you disable DHCHAP, all related configurations are automatically discarded.

```
MDS-9216(config)# feature fcsp
```

**Step 3** Configure a clear text password for this switch. This password will be used by the connecting device.

```
MDS-9216(config)# fcsp dhchap password rtp9216
```

**Step 4** Configures a password for another switch in the fabric that is identified by the switch WWN device name.

```
MDS-9216(config)# fcsp dhchap devicename 20:00:00:05:30:00:38:5e password rtp9509
```

**Step 5** Enable the DHCHAP mode for the required Fibre Channel interface.

**Note** Whenever DHCHAP port mode is changed to a mode other than the Off mode, reauthentication is performed.

```
MDS-9216(config)# interface fc 1/16
MDS-9216(config-if)# fcsp on
```

**Step 6** Verify the protocol security information configured in this switch by displaying the DHCHAP local password database.

```
MDS-9216# show fcsp dhchap database
DHCHAP Local Password:
        Non-device specific password:*******
Other Devices' Passwords:
        Password for device with WWN:20:00:00:05:30:00:38:5e is *******
```

**Step 7** Display the DHCHAP configuration in the Fibre Channel interface.

```
MDS-9216# show fcsp interface fc 1/6
fc1/6
        fcsp authentication mode:SEC_MODE_ON
        Status:Successfully authenticated
```

**Step 8** Repeat these steps on the connecting MDS 9509 Switch.

```
MDS-9509# show wwn switch
Switch WWN is 20:00:00:05:30:00:38:5e
MDS-9509(config)# feature fcsp
MDS-9509(config)# fcsp dhchap password rtp9509
MDS-9509(config)# fcsp dhchap devicename 20:00:00:05:30:00:54:de password rtp9216
MDS-9509(config)# interface fc 4/5
MDS-9509(config-if)# fcsp on
MDS-9509# show fcsp dhchap database
DHCHAP Local Password:
        Non-device specific password:*******
Other Devices' Passwords:
        Password for device with WWN:20:00:00:05:30:00:54:de is *******
MDS-9509# show fcsp interface fc 4/5
Fc4/5
        fcsp authentication mode:SEC_MODE_ON
        Status:Successfully authenticated
```

You have now enabled and configured DHCHAP authentication for the sample setup in Figure 8-2.

# Default Settings

Table 8-2 lists the default settings for all fabric security features in any switch.

*Table 8-2 Default Fabric Security Settings*

| Parameters | Default |
|---|---|
| DHCHAP feature | Disabled |
| DHCHAP hash algorithm | A priority list of MD5 followed by SHA-1 for DHCHAP authentication |
| DHCHAP authentication mode | Auto-passive |
| DHCHAP group default priority exchange order | 0, 4, 1, 2, and 3 respectively |
| DHCHAP timeout value | 30 seconds |

CHAPTER 9

# Configuring Port Security

All switches in the Cisco MDS 9000 Family provide port security features that reject intrusion attempts and report these intrusions to the administrator.

**Note** Port security is supported for Fibre Channel ports and Fibre Channel over Ethernet (FCoE) ports as fc-port-security.

This chapter includes the following sections:



# About Port Security

All switches in the Cisco MDS 9000 Family provide port security features that reject intrusion attempts and report these intrusions to the administrator.

Typically, any Fibre Channel device in a SAN can attach to any SAN switch port and access SAN services based on zone membership. Port security features prevent unauthorized access to a switch port in the Cisco MDS 9000 Family in the following ways:

- Login requests from unauthorized Fibre Channel devices (Nx ports) and switches (xE ports) are rejected.
- All intrusion attempts are reported to the SAN administrator through system messages.
- Configuration distribution uses the CFS infrastructure, and is limited to those switches that are CFS capable. Distribution is disabled by default.
- Configuring the port security policy requires the ENTERPRISE_PKG license (see the *Cisco MDS 9000 Family NX-OS Licensing Guide*).

This section includes the following topics:



# Port Security Enforcement

To enforce port security, configure the devices and switch port interfaces through which each device or switch is connected, and activate the configuration.

- Use the port world wide name (pWWN) or the node world wide name (nWWN) to specify the Nx port connection for each device.
- Use the switch world wide name (sWWN) to specify the xE port connection for each switch.

Each Nx and xE port can be configured to restrict a single port or a range of ports.

Enforcement of port security policies are done on every activation and when the port tries to come up.

The port security feature uses two databases to accept and implement configuration changes.

- Configuration database—All configuration changes are stored in the configuration database.
- Active database—The database currently enforced by the fabric. The port security feature requires all devices connecting to a switch to be part of the port security active database. The software uses this active database to enforce authorization.

# About Auto-Learning

You can instruct the switch to automatically learn (auto-learn) the port security configurations over a specified period. This feature allows any switch in the Cisco MDS 9000 Family to automatically learn about devices and switches that connect to it. Use this feature when you activate the port security feature for the first time as it saves tedious manual configuration for each port. You must configure auto-learning on a per-VSAN basis. If enabled, devices and switches that are allowed to connect to the switch are automatically learned, even if you have not configured any port access.

When auto-learning is enabled, learning happens for the devices or interfaces that were already logged into the switch and the new devices or interfaces that need to be logged in. Learned entries on a port are cleaned up after you shut down that port if auto-learning is still enabled.

Learning does not override the existing configured port security policies. So, for example, if an interface is configured to allow a specific pWWN, then auto-learning will not add a new entry to allow any other pWWN on that interface. All other pWWNs will be blocked even in auto-learning mode.

No entries are learned for a port in the shutdown state.

When you activate the port security feature, auto-learning is also automatically enabled.

**Note** If you activate port security feature, auto-learning gets enabled by default. You cannot re-activate port security until auto-learning is disabled or deactivate and activate again.

## Port Security Activation

By default, the port security feature is not activated in any switch in the Cisco MDS 9000 Family.

By activating the port security feature, the following apply:

- Auto-learning is also automatically enabled, which means:
  - From this point, auto-learning happens for the devices or interfaces that were already logged into the switch and also for the new devices will login in future.
  - You cannot activate the database until you disable auto-learning.
- All the devices that are already logged in are learned and are added to the active database.
- All entries in the configured database are copied to the active database.

After the database is activated, subsequent device login is subject to the activated port bound WWN pairs, excluding the auto-learned entries. You must disable auto-learning before the auto-learned entries become activated.

When you activate the port security feature, auto-learning is also automatically enabled. You can choose to activate the port security feature and disable auto-learning.

**Tip** If a port is shut down because of a denied login attempt, and you subsequently configure the database to allow that login, the port does not come up automatically. You must explicitly issue a **no shutdown** CLI command to bring that port back online.

# Port Security Configuration

The steps to configure port security depend on which features you are using. Auto-learning works differently if you are using CFS distribution.

This section includes the following topics:

# Configuring Port Security with Auto-Learning and CFS Distribution

To configure port security, using auto-learning and CFS distribution, follow these steps:

**Step 1** Enable port security. See the "Enabling Port Security" section on page 9-229.

**Step 2** Enable CFS distribution. See the "Enabling Distribution" section on page 9-236.

**Step 3** Activate port security on each VSAN. This turns on auto-learning by default. See the "Activating Port Security" section on page 9-230.

**Step 4** Issue a CFS commit to copy this configuration to all switches in the fabric. See the "Committing the Changes" section on page 9-237. At this point, all switches are activated, and auto-learning.

**Step 5** Wait until all switches and all hosts are automatically learned.

**Step 6** Disable auto-learn on each VSAN. See the"Disabling Auto-learning" section on page 9-232.

**Step 7** Issue a CFS commit to copy this configuration to all switches in the fabric. See the "Committing the Changes" section on page 9-237. At this point, the auto-learned entries from every switch are combined into a static active database that is distributed to all switches.

**Step 8** Copy the active database to the configure database on each VSAN. See the "Copying the Port Security Database" section on page 9-240.

**Step 9** Issue a CFS commit to copy this configuration to all switches in the fabric. See the "Committing the Changes" section on page 9-237. This ensures that the configure database is the same on all switches in the fabric.

**Step 10** Copy the running configuration to the startup configuration, using the fabric option. This saves the port security configure database to the startup configuration on all switches in the fabric.

# Configuring Port Security with Auto-Learning without CFS

To configure port security using auto-learning without CFS, follow these steps:

**Step 1** Enable port security. See the "Enabling Port Security" section on page 9-229.

**Step 2** Activate port security on each VSAN. This turns on auto-learning by default. See the "Activating Port Security" section on page 9-230.

**Step 3** Wait until all switches and all hosts are automatically learned.

**Step 4** Disable auto-learn on each VSAN. See the "Disabling Auto-learning" section on page 9-232.

**Step 5** Copy the active database to the configure database on each VSAN. See the "Copying the Port Security Database" section on page 9-240.

**Step 6** Copy the running configuration to the startup configuration This saves the port security configure database to the startup configuration.

**Step 7** Repeat Step 1 through Step 6 for all switches in the fabric.

## Configuring Port Security with Manual Database Configuration

To configure port security and manually configure the port security database, follow these steps:

**Step 1** Enable port security. See the "Enabling Port Security" section on page 9-229.

**Step 2** Manually configure all port security entries into the configure database on each VSAN. See the "Port Security Manual Configuration" section on page 9-234.

**Step 3** Activate port security on each VSAN. This turns on auto-learning by default. See the "Activating Port Security" section on page 9-230.

**Step 4** Disable auto-learn on each VSAN. See the "Disabling Auto-learning" section on page 9-232.

**Step 5** Copy the running configuration to the startup configuration This saves the port security configure database to the startup configuration.

**Step 6** Repeat Step 1 through Step 5 for all switches in the fabric.

# Enabling Port Security

By default, the port security feature is disabled in all switches in the Cisco MDS 9000 Family.

To enable port security, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# feature port-security` | Enables port security on that switch. |
| | `switch(config)# no feature port-security` | Disables (default) port security on that switch. |

# Port Security Activation

This section includes the following topics:

- Activating Port Security, page 9-230
- Database Activation Rejection, page 9-230
- Forcing Port Security Activation, page 9-230

# Activating Port Security

To activate the port security feature, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# port-security activate vsan 1` | Activates the port security database for the specified VSAN, and automatically enables auto-learning. |
| | `switch(config)# port-security activate vsan 1 no-auto-learn` | Activates the port security database for the specified VSAN, and disables auto-learning. |
| | `switch(config)# no port-security activate vsan 1` | Deactivates the port security database for the specified VSAN, and automatically disables auto-learning. |

**Note** If required, you can disable auto-learning (see the "Disabling Auto-learning" section on page 9-232).

## Database Activation Rejection

Database activation is rejected in the following cases:

- Missing or conflicting entries exist in the configuration database but not in the active database.
- The auto-learning feature was enabled before the activation. To reactivate a database in this state, disable auto-learning.
- The exact security is not configured for each PortChannel member.
- The configured database is empty but the active database is not.

If the database activation is rejected due to one or more conflicts listed in the previous section, you may decide to proceed by forcing the port security activation.

## Forcing Port Security Activation

If the port security activation request is rejected, you can force the activation.

**Note** An activation using the **force** option can log out existing devices if they violate the active database.

You can view missing or conflicting entries using the **port-security database diff active vsan** command in EXEC mode.

To forcefully activate the port security database, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# port-security activate vsan 1 force` | Forces the VSAN 1 port security database to activate despite conflicts. |

## Database Reactivation

To reactivate the port security database, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# no port-security auto-learn vsan 1` | Disables auto-learning and stops the switch from learning about new devices accessing the switch. Enforces the database contents based on the devices learned up to this point. |
| **Step 3** | `switch(config)# exit`<br>`switch# port-security database copy vsan 1` | Copies from the active to the configured database. |
| **Step 4** | `switch# config t`<br>`switch(config)# port-security activate vsan 1` | Activates the port security database for the specified VSAN, and automatically enables auto-learning. |

**Tip** If auto-learning is enabled, and you cannot activate the database, you will not be allowed to proceed without the **force** option until you disable auto-learning.

# Auto-learning

This section contains the following topics:



## About Enabling Auto-learning

The state of the auto-learning configuration depends on the state of the port security feature:

- If the port security feature is not activated, auto-learning is disabled by default.
- If the port security feature is activated, auto-learning is enabled by default (unless you explicitly disabled this option).

**Tip** If auto-learning is enabled on a VSAN, you can only activate the database for that VSAN by using the **force** option.

# Enabling Auto-learning

To enable auto-learning, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# port-security auto-learn vsan 1` | Enables auto-learning so the switch can learn about any device that is allowed to access VSAN 1. These devices are logged in the port security active database. |

# Disabling Auto-learning

To disable auto-learning, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# no port-security auto-learn vsan 1` | Disables auto-learning and stops the switch from learning about new devices accessing the switch. Enforces the database contents based on the devices learned up to this point. |

# Auto-learning Device Authorization

Table 9-1 summarizes the authorized connection conditions for device requests.

*Table 9-1 Authorized Auto-learning Device Requests*

| Condition | Device (pWWN, nWWN, sWWN) | Requests Connection to | Authorization |
|---|---|---|---|
| 1 | Configured with one or more switch ports | A configured switch port | Permitted |
| 2 | | Any other switch port | Denied |
| 3 | Not configured | A switch port that is not configured | Permitted if auto-learning enabled |
| 4 | | | Denied if auto-learning disabled |
| 5 | Configured or not configured | A switch port that allows any device | Permitted |

*Table 9-1 Authorized Auto-learning Device Requests (continued)*

| Condition | Device (pWWN, nWWN, sWWN) | Requests Connection to | Authorization |
|---|---|---|---|
| 6 | Configured to log in to any switch port | Any port on the switch | Permitted |
| 7 | Not configured | A port configured with some other device | Denied |

# Authorization Scenarios

Assume that the port security feature is activated and the following conditions are specified in the active database:

- A pWWN (P1) is allowed access through interface fc1/1 (F1).
- A pWWN (P2) is allowed access through interface fc1/1 (F1).
- A nWWN (N1) is allowed access through interface fc1/2 (F2).
- Any WWN is allowed access through interface fc1/3 (F3).
- A nWWN (N3) is allowed access through any interface.
- A pWWN (P3) is allowed access through interface fc1/4 (F4).
- A sWWN (S1) is allowed access through interface fc1/10-13 (F10 to F13).
- A pWWN (P10) is allowed access through interface fc1/11 (F11).

Table 9-2 summarizes the port security authorization results for this active database. The conditions listed refer to the conditions from Table 9-1.

*Table 9-2 Authorization Results for Scenario*

| Device Connection Request | Authorization | Condition | Reason |
|---|---|---|---|
| P1, N2, F1 | Permitted | 1 | No conflict. |
| P2, N2, F1 | Permitted | 1 | No conflict. |
| P3, N2, F1 | Denied | 2 | F1 is bound to P1/P2. |
| P1, N3, F1 | Permitted | 6 | Wildcard match for N3. |
| P1, N1, F3 | Permitted | 5 | Wildcard match for F3. |
| P1, N4, F5 | Denied | 2 | P1 is bound to F1. |
| P5, N1, F5 | Denied | 2 | N1 is only allowed on F2. |
| P3, N3, F4 | Permitted | 1 | No conflict. |
| S1, F10 | Permitted | 1 | No conflict. |
| S2, F11 | Denied | 7 | P10 is bound to F11. |
| P4, N4, F5 (auto-learning on) | Permitted | 3 | No conflict. |
| P4, N4, F5(auto-learning off) | Denied | 4 | No match. |
| S3, F5 (auto-learning on) | Permitted | 3 | No conflict. |
| S3, F5 (auto-learning off) | Denied | 4 | No match. |

*Table 9-2 Authorization Results for Scenario (continued)*

| Device Connection Request | Authorization | Condition | Reason |
|---|---|---|---|
| P1, N1, F6 (auto-learning on) | Denied | 2 | P1 is bound to F1. |
| P5, N5, F1 (auto-learning on) | Denied | 7 | Only P1 and P2 bound to F1. |
| S3, F4 (auto-learning on) | Denied | 7 | P3 paired with F4. |
| S1, F3 (auto-learning on) | Permitted | 5 | No conflict. |
| P5, N3, F3 | Permitted | 6 | Wildcard ( * ) match for F3 and N3. |
| P7, N3, F9 | Permitted | 6 | Wildcard ( * ) match for N3. |

# Port Security Manual Configuration

To configure port security on any switch in the Cisco MDS 9000 Family, follow these steps:

**Step 1** Identify the WWN of the ports that need to be secured.

**Step 2** Secure the fWWN to an authorized nWWN or pWWN.

**Step 3** Activate the port security database.

**Step 4** Verify your configuration.

This section includes the following topics:



## About WWN Identification

If you decide to manually configure port security, be sure to adhere to the following guidelines:

- Identify switch ports by the interface or by the fWWN.
- Identify devices by the pWWN or by the nWWN.
- If an Nx port is allowed to log in to SAN switch port Fx, then that Nx port can only log in through the specified Fx port.
- If an Nx port's nWWN is bound to an Fx port WWN, then all pWWNs in the Nx port are implicitly paired with the Fx port.
- TE port checking is done on each VSAN in the allowed VSAN list of the trunk port.
- All PortChannel xE ports must be configured with the same set of WWNs in the same PortChannel.
- E port security is implemented in the port VSAN of the E port. In this case the sWWN is used to secure authorization checks.
- Once activated, the config database can be modified without any effect on the active database.

- By saving the running configuration, you save the configuration database and activated entries in the active database. Learned entries in the active database are not saved.

# Adding Authorized Port Pairs

To add authorized port pairs for port security, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# port-security database vsan 1`<br>`switch(config-port-security)#` | Enters the port security database mode for the specified VSAN. |
| | `switch(config)# no port-security database vsan 1`<br>`switch(config)#` | Deletes the port security configuration database from the specified VSAN. |
| **Step 3** | `switch(config-port-security)# swwn 20:01:33:11:00:2a:4a:66 interface port-channel 5` | Configures the specified sWWN to only log in through PortChannel 5. |
| | `switch(config-port-security)# any-wwn interface fc1/1 - fc1/8` | Configures any WWN to log in through the specified interfaces. |
| | `switch(config-port-security)# pwwn 20:11:00:33:11:00:2a:4a fwwn 20:81:00:44:22:00:4a:9e` | Configures the specified pWWN to only log in through the specified fWWN. |
| | `switch(config-port-security)# no pwwn 20:11:00:33:11:00:2a:4a fwwn 20:81:00:44:22:00:4a:9e` | Deletes the specified pWWN configured in the previous step. |
| | `switch(config-port-security)# nwwn 26:33:22:00:55:05:3d:4c fwwn 20:81:00:44:22:00:4a:9e` | Configures the specified nWWN to log in through the specified fWWN. |
| | `switch(config-port-security)# pwwn 20:11:33:11:00:2a:4a:66` | Configures the specified pWWN to log in through any port in the fabric. |
| | `switch(config-port-security)# pwwn 20:11:33:11:00:2a:4a:66 swwn 20:00:00:0c:85:90:3e:80` | Configures the specified pWWN to log in through any interface in the specified switch. |
| | `switch(config-port-security)# pwwn 20:11:33:11:00:2a:4a:66 swwn 20:00:00:0c:85:90:3e:80 interface fc3/1` | Configures the specified pWWN to log in through the specified interface in the specified switch |
| | `switch(config-port-security)# any-wwn interface fc3/1` | Configures any WWN to log in through the specified interface in any switch. |
| | `switch(config-port-security)# no any-wwn interface fc2/1` | Deletes the wildcard configured in the previous step. |

After identifying the WWN pairs that need to be bound, add those pairs to the port security database.

**Tip** Remote switch binding can be specified at the local switch. To specify the remote interfaces, you can use either the fWWN or sWWN-interface combination.

# Port Security Configuration Distribution

The port security feature uses the Cisco Fabric Services (CFS) infrastructure to enable efficient database management, provide a single point of configuration for the entire fabric in the VSAN, and enforce the port security policies throughout the fabric.

This section includes the following topics:



## Enabling Distribution

To enable the port security distribution, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# port-security distribute` | Enables distribution. |
| | `switch(config)# no port-security distribute` | Disables distribution. |

For example, if you activate port security, follow up by disabling auto-learning, and commit the changes in the pending database, then the net result of your actions is the same as issuing a **port-security activate vsan** *vsan-id* **no-auto-learn** command.

All the configurations performed in distributed mode are stored in a pending (temporary) database. If you modify the configuration, you need to commit or discard the pending database changes to the configurations. The fabric remains locked during this period. Changes to the pending database are not reflected in the configurations until you commit the changes.

**Note** Port activation or deactivation and auto-learning enable or disable do not take effect until after a CFS commit if CFS distribution is enabled. Always follow any one of these operations with a CFS commit to ensure proper configuration. See the "Activation and Auto-learning Configuration Distribution" section on page 9-237.

**Tip** In this case, we recommend that you perform a commit at the end of each operation: after you activate port security and after you enable auto learning.

## Locking the Fabric

The first action that modifies the existing configuration creates the pending database and locks the feature in the VSAN. After you lock the fabric, the following situations apply:

- No other user can make any configuration changes to this feature.

- A copy of the configuration database becomes the pending database.

To display the CFS lock information, use the **show cfs lock** command. For more information, see the *Cisco MDS 9000 Family Command Reference*.

# Committing the Changes

If you commit the changes made to the configurations, the configurations in the pending database are distributed to other switches. On a successful commit, the configuration change is applied throughout the fabric and the lock is released.

To commit the port security configuration changes for the specified VSAN, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# port-security commit vsan 3` | Commits the port security changes in the specified VSAN. |

# Discarding the Changes

If you discard (abort) the changes made to the pending database, the configuration remains unaffected and the lock is released.

To display the CFS lock information, use the **show cfs lock** command. For more information, see the Cisco MDS 9000 Family Command Reference.

To discard the port security configuration changes for the specified VSAN, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# port-security abort vsan 5` | Discards the port security changes in the specified VSAN and clears the pending configuration database. |

# Activation and Auto-learning Configuration Distribution

Activation and auto-learning configurations in distributed mode are remembered as actions to be performed when you commit the changes in the pending database.

Learned entries are temporary and do not have any role in determining if a login is authorized or not. As such, learned entries do not participate in distribution. When you disable learning and commit the changes in the pending database, the learned entries become static entries in the active database and are distributed to all switches in the fabric. After the commit, the active database on all switches is identical.

If the pending database contains more than one activation and auto-learning configuration when you commit the changes, then the activation and auto-learning changes are consolidated and the behavior may change (see Table 9-3).

*Table 9-3 Scenarios for Activation and Auto- learning Configurations in Distributed Mode*

| Scenario | Actions | Distribution = OFF | Distribution = ON |
|---|---|---|---|
| A and B exist in the configuration database, activation is not done and devices C,D are logged in. | 1. You activate the port security database and enable auto-learning. | configuration database = {A,B}<br>active database = {A,B, C[1], D*} | configuration database = {A,B}<br>active database = {null}<br>pending database = {A,B + activation to be enabled} |
| | 2. A new entry E is added to the configuration database. | configuration database = {A,B, E}<br>active database = {A,B, C*, D*} | configuration database = {A,B}<br>active database = {null}<br>pending database = {A,B, E + activation to be enabled} |
| | 3. You issue a commit. | Not applicable | configuration database = {A,B, E}<br>active database = {A,B, E, C*, D*}<br>pending database = empty |
| A and B exist in the configuration database, activation is not done and devices C,D are logged in. | 1. You activate the port security database and enable auto-learning. | configuration database = {A,B}<br>active database = {A,B, C*, D*} | configuration database = {A,B}<br>active database = {null}<br>pending database = {A,B + activation to be enabled} |
| | 2. You disable learning. | configuration database = {A,B}<br>active database = {A,B, C, D} | configuration database = {A,B}<br>active database = {null}<br>pending database = {A,B + activation to be enabled +<br>learning to be disabled} |
| | 3. You issue a commit. | Not applicable | configuration database = {A,B}<br>active database = {A,B} and devices C and D are logged out. This is equal to an activation with auto-learning disabled.<br>pending database = empty |

1. The * (asterisk) indicates learned entries.

**Tip** In this case, we recommend that you perform a commit at the end of each operation: after you activate port security and after you enable auto-learning.

# Database Merge Guidelines

A database merge refers to a union of the configuration database and static (unlearned) entries in the active database.

When merging the database between two fabrics, follow these guidelines:

- Verify that the activation status and the auto-learning status is the same in both fabrics.

- Verify that the combined number of configurations for each VSAN in both databases does not exceed 2 K.

**Caution** If you do not follow these two conditions, the merge will fail. The next distribution will forcefully synchronize the databases and the activation states in the fabric.

# Database Interaction

Table 9-4 lists the differences and interaction between the active and configuration databases.

*Table 9-4 Active and Configuration Port Security Databases*

| Active Database | Configuration Database |
|---|---|
| Read-only. | Read-write. |
| Saving the configuration only saves the activated entries. Learned entries are not saved. | Saving the configuration saves all the entries in the configuration database. |
| Once activated, all devices that have already logged into the VSAN are also learned and added to the active database. | Once activated, the configuration database can be modified without any effect on the active database. |
| You can overwrite the active database with the configured database by activating the port security database. Forcing an activation may violate the entries already configured in the active database. | You can overwrite the configuration database with the active database. |

**Note** You can overwrite the configuration database with the active database using the **port-security database copy vsan** command. The **port-security database diff active vsa**n command in EXEC mode lists the differences between the active database and the configuration database.

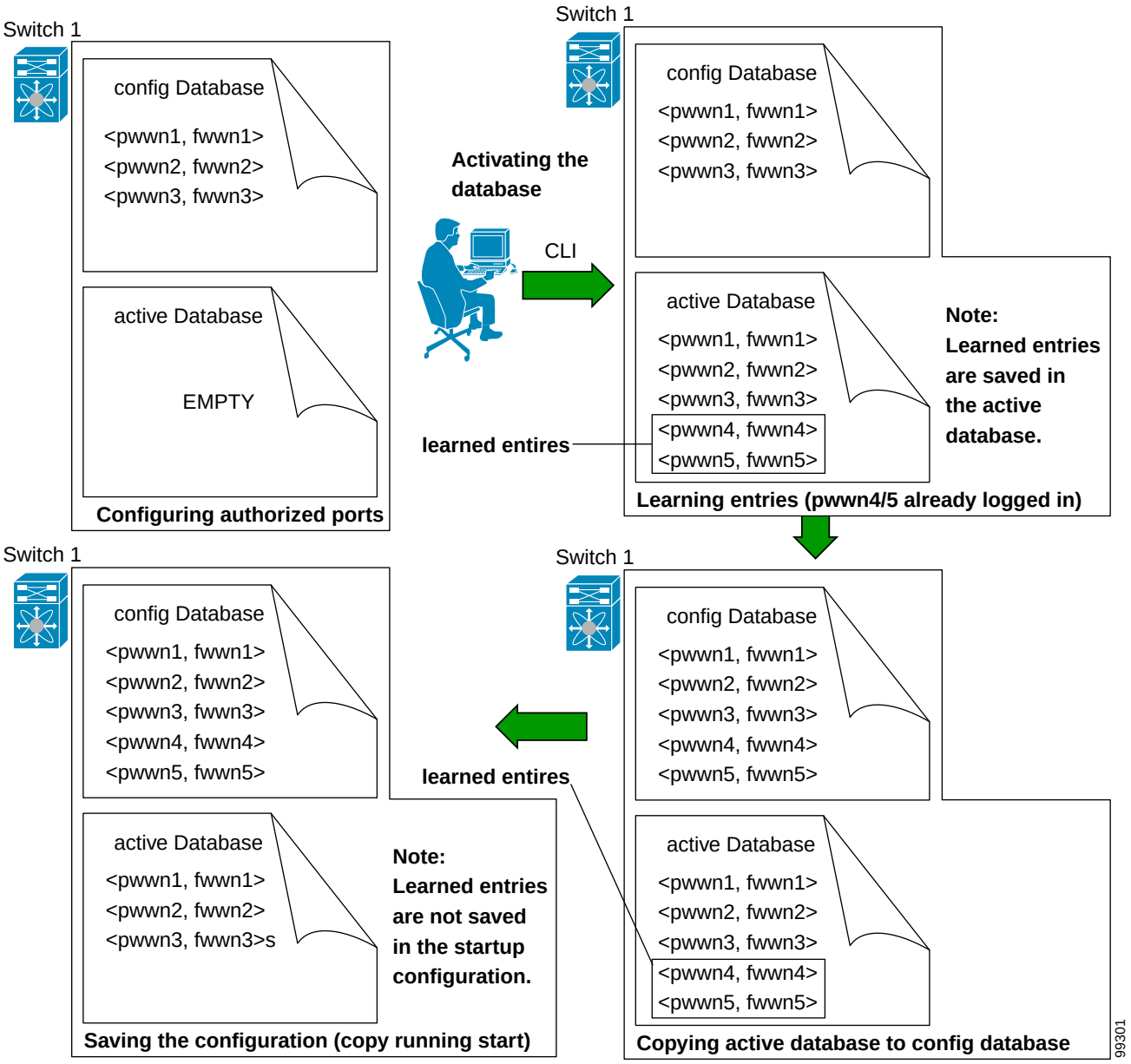
This section includes the following topics:



# Database Scenarios

Figure 9-1 depicts various scenarios to depict the active database and the configuration database status based on port security configurations.

*Figure 9-1 Port Security Database Scenarios*




# Copying the Port Security Database

Use the **port-security database copy vsan** command to copy from the active to the configured database. If the active database is empty, this command is not accepted.

```
switch# port-security database copy vsan 1
```

Use the **port-security database diff active vsan** command to view the differences between the active database and the configuration database. This command can be used when resolving conflicts.

```
switch# port-security database diff active vsan 1
```

Use the **port-security database diff config vsan** command to obtain information on the differences between the configuration database and the active database.

```
switch# port-security database diff config vsan 1
```

**Tip** We recommend that you issue the **port-security database copy vsan** command after disabling auto-learning. This action will ensure that the configuration database is in sync with the active database. If distribution is enabled, this command creates a temporary copy (and consequently a fabric lock) of the configuration database. If you lock the fabric, you need to commit the changes to the configuration databases in all the switches.

# Deleting the Port Security Database

**Tip** If the distribution is enabled, the deletion creates a copy of the database. An explicit **port-security commit** command is required to actually delete the database.

Use the **no port-security database vsan** command in configuration mode to delete the configured database for a specified VSAN

```
switch(config)# no port-security database vsan 1
```

# Cleaning the Port Security Database

Use the **clear port-security statistics vsan** command to clear all existing statistics from the port security database for a specified VSAN.

```
switch# clear port-security statistics vsan 1
```

Use the **clear port-security database auto-learn interface** command to clear any learned entries in the active database for a specified interface within a VSAN.

```
switch# clear port-security database auto-learn interface fc1/1 vsan 1
```

Use the **clear port-security database auto-learn vsan** command to clear any learned entries in the active database for the entire VSAN.

```
switch# clear port-security database auto-learn vsan 1
```

**Note** The **clear port-security database auto-learn** and **clear port-security statistics** commands are only relevant to the local switch and do not acquire locks. Also, learned entries are only local to the switch and do not participate in distribution.

Use the **port-security clear vsan** command to clear the pending session in the VSAN from any switch in the VSAN.

```
switch# clear port-security session vsan 5
```

# Displaying Port Security Configuration

The **show port-security database** commands display the configured port security information (see Examples 9-1 to 9-11).

***Example 9-1 Displays the Contents of the Port Security Configuration Database***

```
switch# show port-security database
--------------------------------------------------------------------------------
VSAN    Logging-in Entity               Logging-in Point       (Interface)
--------------------------------------------------------------------------------
1       21:00:00:e0:8b:06:d9:1d(pwwn)  20:0d:00:05:30:00:95:de(fc1/13)
1       50:06:04:82:bc:01:c3:84(pwwn)  20:0c:00:05:30:00:95:de(fc1/12)
2       20:00:00:05:30:00:95:df(swwn)  20:0c:00:05:30:00:95:de(port-channel 128)
3       20:00:00:05:30:00:95:de(swwn)  20:01:00:05:30:00:95:de(fc1/1)
[Total 4 entries]
```

You can optionally specify a fWWN and a VSAN, or an interface and a VSAN in the **show port-security** command to view the output of the activated port security (see Example 9-2).

***Example 9-2 Displays the Port Security Configuration Database in VSAN 1***

```
switch# show port-security database vsan 1
--------------------------------------------------------------------------------
Vsan   Logging-in Entity              Logging-in Point        (Interface)
--------------------------------------------------------------------------------
1                  *                  20:85:00:44:22:00:4a:9e (fc3/5)
1      20:11:00:33:11:00:2a:4a(pwwn)  20:81:00:44:22:00:4a:9e (fc3/1)
[Total 2 entries]
```

***Example 9-3 Displays the Activated Database***

```
switch# show port-security database active
--------------------------------------------------------------------------------------
VSAN    Logging-in Entity               Logging-in Point     (Interface)         Learnt
--------------------------------------------------------------------------------------
1       21:00:00:e0:8b:06:d9:1d(pwwn)   20:0d:00:05:30:00:95:de(fc1/13)           Yes
1       50:06:04:82:bc:01:c3:84(pwwn)   20:0c:00:05:30:00:95:de(fc1/12)           Yes
2       20:00:00:05:30:00:95:df(swwn)   20:0c:00:05:30:00:95:de(port-channel 128) Yes
3       20:00:00:05:30:00:95:de(swwn)   20:01:00:05:30:00:95:de(fc1/1)
[Total 4 entries]
```

***Example 9-4 Displays the Contents of the Temporary Configuration Database***

```
switch# show port-security pending vsan 1
Session Context for VSAN 1
---------------------------
Activation Status: Active
Auto Learn Status: On
Force activate: No
Config db modified: Yes
Activation done: Yes
Session owner: admin(2)
Session database:
--------------------------------------------------------------------------------
VSAN Logging-in Entity Logging-in Point (Interface)
--------------------------------------------------------------------------------
```

```
1 20:11:00:33:22:00:2a:4a(pwwn) 20:41:00:05:30:00:4a:1e(fc2/1)
[Total 1 entries]
```

***Example 9-5 Displays the Difference Between the Temporary Configuration Database and the Configuration Database***

```
switch# show port-security pending-diff vsan 1
Session Diff for VSAN: 1
------------------------
Database will be activated
Learning will be turned ON
Database Diff:
+pwwn 20:11:00:33:22:00:2a:4a fwwn 20:41:00:05:30:00:4a:1e
```

The access information for each port can be individually displayed. If you specify the fWWN or interface options, all devices that are paired in the active database (at that point) with the given fWWN or the interface are displayed (see Examples 9-6 to 9-8).

***Example 9-6 Displays the Wildcard fWWN Port Security in VSAN 1***

```
switch# show port-security database fwwn 20:85:00:44:22:00:4a:9e vsan 1
Any port can login thru' this fwwn
```

***Example 9-7 Displays the Configured fWWN Port Security in VSAN 1***

```
switch# show port-security database fwwn 20:01:00:05:30:00:95:de vsan 1
20:00:00:0c:88:00:4a:e2(swwn)
```

***Example 9-8 Displays the Interface Port Information in VSAN 2***

```
switch# show port-security database interface fc 1/1 vsan 2
20:00:00:0c:88:00:4a:e2(swwn)
```

The port security statistics are constantly updated and available at any time (see Example 9-9).

***Example 9-9 Displays the Port Security Statistics***

```
switch# show port-security statistics
Statistics For VSAN: 1
-----------------------
Number of pWWN permit: 2
Number of nWWN permit: 2
Number of sWWN permit: 2
Number of pWWN deny  : 0
Number of nWWN deny  : 0
Number of sWWN deny  : 0

Total Logins permitted  : 4
Total Logins denied     : 0
Statistics For VSAN: 2
-----------------------
Number of pWWN permit: 0
Number of nWWN permit: 0
Number of sWWN permit: 2
Number of pWWN deny  : 0
Number of nWWN deny  : 0
```

```
Number of sWWN deny  : 0
...
```

To verify the status of the active database and the auto-learning configuration, use the **show port-security status** command (see Example 9-10).

***Example 9-10 Displays the Port Security Status***

```
switch# show port-security status
Fabric Distribution Enabled
VSAN 1 :No Active database, learning is disabled, Session Lock Taken
VSAN 2 :No Active database, learning is disabled, Session Lock Taken
...
```

The **show port-security** command displays the previous 100 violations by default (see Example 9-11).

***Example 9-11 Displays the Violations in the Port Security Database***

```
switch# show port-security violations

-------------------------------------------------------------------------------------
VSAN    Interface       Logging-in Entity               Last-Time          [Repeat count]
-------------------------------------------------------------------------------------
1       fc1/13          21:00:00:e0:8b:06:d9:1d(pwwn)   Jul  9 08:32:20 2003    [20]
                        20:00:00:e0:8b:06:d9:1d(nwwn)
1       fc1/12          50:06:04:82:bc:01:c3:84(pwwn)   Jul  9 08:32:20 2003    [1]
                        50:06:04:82:bc:01:c3:84(nwwn)
2       port-channel 1  20:00:00:05:30:00:95:de(swwn)   Jul  9 08:32:40 2003    [1]
[Total 2 entries]
```

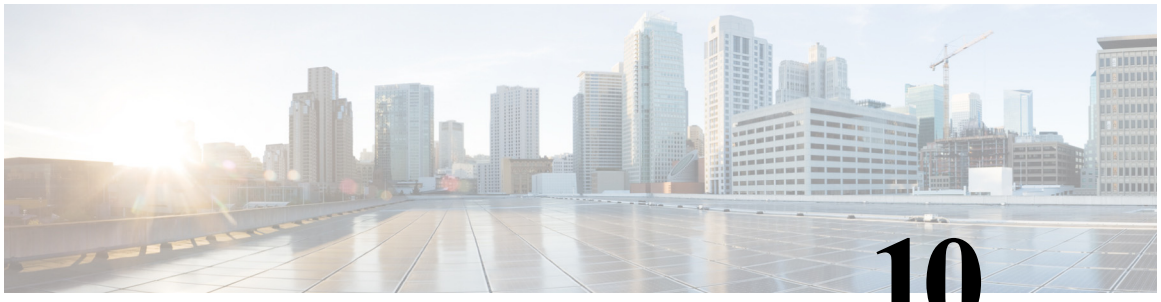
The **show port-security** command issued with the **last** *number* option displays only the specified number of entries that appear first.

# Default Settings

Table 9-5 lists the default settings for all port security features in any switch.

***Table 9-5 Default Security Settings***

| Parameters | Default |
|---|---|
| Auto-learn | Enabled if port security is enabled. |
| Port security | Disabled |
| Distribution | Disabled.<br>**Note** Enabling distribution enables it on all VSANs in the switch. |



CHAPTER 10

# Configuring Fibre Channel Common Transport Management Security

This chapter describes the Fibre Channel Common Transport (FC-CT) Management Security feature for Cisco MDS 9000 Series switches.

## About Fibre Channel Common Transport

With the FC-CT management security feature, you can configure the network in such a manner that only a storage administrator or a network administrator can send queries to a switch and access information such as devices that are logged in devices in the fabric, switches in the fabric, how they are connected, how many ports each switch has and where each port is connected, configured zone information and privilege to add or delete zone and zone sets, and host bus adapter (HBA) details of all the hosts connected in the fabric.

**Note** In Cisco MDS NX-OS Release 6.2(9), the FC management feature is disabled by default. To enable FC management feature, use the **fc-management enable** command.

You can configure which pWWNs can send FC-CT management query and modify request to the management server. When any of the modules, such as a zone server, unzoned Fibre Channel name server (FCNS), or Fabric Configuration Server (FCS) receives an FC-CT management query, they perform a read operation on the FC-management database. If device is found in FC-management database, a reply is sent according to the permissions granted. If the device is not found in the FC-management database, each module sends a reject. If FC-management is disabled, each module processes each management query.

## Configuration Guidelines

The FC-management security feature has the following configuration guidelines:

- When the FC-management security feature is enabled on a Cisco MDS switch, all management queries to the server are rejected unless the port world-wide name (pWWN) of the device that is sending management queries is added to FC-management database.

- When you enable FC Management, FC-CT management server queries from N_Port Virtualization (NPV) switches to N_Port Identifier Virtualization (NPIV) switches are rejected. We recommend that you add the switch world-wide name (sWWN) of the NPV switch to the FC management database of the NPIV switch after enabling the FC-management security feature.

# Configuring the Fibre Channel Common Transport Query

To configure the FC-CT management security, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config terminal` | Enters configuration mode. |
| **Step 2** | `switch(config)# fc-management enable`<br>`switch(config)#` | Enables the FC-CT management security. |
| **Step 3** | `switch(config)# fc-management database vsan 1` | Configures the FC-CT management Security database. |
| **Step 4** | `switch(config-fc-mgmt)# pwwn 1:1:1:1:1:1:1:1 feature all operation both` | Adds the pWWN to the FC management database. You also can use these optional keywords when configuring the **pwwn** command:<br>• **fcs**— Enables or disables FC-CT query for fabric conf-server.<br>• **fdmi**—Enables or disables FC-CT query for FDMI.<br>• **unzoned-ns**—Enables or disables FC-CT query for unzoned name-server.<br>• **zone**—Enables or disables FC-CT query for zone-server. |
| **Step 5** | `switch# show fc-managment database` | Displays the configured FC-CT management information. |

# Verifying Fibre Channel Common Transport Management Security

The **show fc-management database** command displays the configured FC-CT management security feature information, see example 10-1.

***Example 10-1 Displays the Contents of the Fibre Channel Common Transport Query***

```
switch# show fc-management database
-----------------------------------------------------------------
VSAN PWWN FC-CT Permissions per FC services
-----------------------------------------------------------------
1 01:01:01:01:01:01:01:01 Zone(RW), Unzoned-NS(RW), FCS(RW), FDMI(RW)
1 02:02:02:02:02:02:02:02 Zone(R), Unzoned-NS(R), FCS(R), FDMI(R)
1 03:03:03:03:03:03:03:03 Zone(W), Unzoned-NS(W), FCS(W), FDMI(W)
-----------------------------------------------------------------
Total 3 entries
switch#
```

To verify the if the FC-management security feature is enabled or not, use the **show fc-management status** command:

```
switch# show fc-management status
Mgmt Security Disabled
switch#
```

# Default Settings

Table 10-1 lists the default settings for the FC management security feature in a Cisco MDS 9000 Family switch.

*Table 10-1 Default FC Management Settings*

| Parameters | Default |
|---|---|
| FC-management | Disabled |


# Configuring Fabric Binding

This chapter describes the fabric binding feature provided in the Cisco MDS 9000 Family of directors and switches. It includes the following sections:



## About Fabric Binding

The fabric binding feature ensures ISLs are only enabled between specified switches in the fabric binding configuration. Fabric binding is configured on a per-VSAN basis.

This feature helps prevent unauthorized switches from joining the fabric or disrupting current fabric operations. It uses the Exchange Fabric Membership Data (EFMD) protocol to ensure that the list of authorized switches is identical in all switches in the fabric.

This section has the following topics:



### Licensing Requirements

Fabric binding requires that you install either the MAINFRAME_PKG license or the ENTERPRISE_PKG license on your switch.

See the *Cisco MDS 9000 Family NX-OS Licensing Guide* for more information on license feature support and installation.

### Port Security Versus Fabric Binding

Port security and fabric binding are two independent features that can be configured to complement each other. Table 11-1 compares the two features.

*Table 11-1 Fabric Binding and Port Security Comparison*

| Fabric Binding | Port Security |
|---|---|
| Uses a set of sWWNs and a persistent domain ID. | Uses pWWNs/nWWNs or fWWNs/sWWNs. |
| Binds the fabric at the switch level. | Binds devices at the interface level. |
| Authorizes only the configured sWWN stored in the fabric binding database to participate in the fabric. | Allows a preconfigured set of Fibre Channel devices to logically connect to a SAN ports. The switch port, identified by a WWN or interface number, connects to a Fibre Channel device (a host or another switch), also identified by a WWN. By binding these two devices, you lock these two ports into a group (or list). |
| Requires activation on a per VSAN basis. | Requires activation on a per VSAN basis. |
| Allows specific user-defined switches that are allowed to connect to the fabric, regardless of the physical port to which the peer switch is connected. | Allows specific user-defined physical ports to which another device can connect. |
| Does not learn about switches that are logging in. | Learns about switches or devices that are logging in if learning mode is enabled. |
| Cannot be distributed by CFS and must be configured manually on each switch in the fabric. | Can be distributed by CFS. |

Port-level checking for xE ports is as follows:

- The switch login uses both port security binding and fabric binding for a given VSAN.
- Binding checks are performed on the port VSAN as follows:
  - E port security binding check on port VSAN
  - TE port security binding check on each allowed VSAN

While port security complements fabric binding, they are independent features and can be enabled or disabled separately.

# Fabric Binding Enforcement

To enforce fabric binding, configure the switch world wide name (sWWN) to specify the xE port connection for each switch. Enforcement of fabric binding policies are done on every activation and when the port tries to come up. In a FICON VSAN, the fabric binding feature requires all sWWNs connected to a switch and their persistent domain IDs to be part of the fabric binding active database. In a Fibre Channel VSAN, only the sWWN is required; the domain ID is optional.

**Note** All switches in a Fibre Channel VSAN using fabric binding must be running Cisco MDS SAN-OS Release 3.0(1) and NX-OS Release 4.1(1b) or later.

# Fabric Binding Configuration

To configure fabric binding in each switch in the fabric, follow these steps:

**Step 1** Enable the fabric configuration feature.

**Step 2** Configure a list of sWWNs and their corresponding domain IDs for devices that are allowed to access the fabric.

**Step 3** Activate the fabric binding database.

**Step 4** Copy the fabric binding active database to the fabric binding config database.

**Step 5** Save the fabric binding configuration.

**Step 6** Verify the fabric binding configuration.

## Enabling Fabric Binding

The fabric binding feature must be enabled in each switch in the fabric that participates in the fabric binding. By default, this feature is disabled in all switches in the Cisco MDS 9000 Family. The configuration and verification commands for the fabric binding feature are only available when fabric binding is enabled on a switch. When you disable this configuration, all related configurations are automatically discarded.

To enable fabric binding on any participating switch, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# feature fabric-binding` | Enables fabric binding on that switch. |
| | `switch(config)# no feature fabric-binding` | Disables (default) fabric binding on that switch. |

View the status of the fabric binding feature of a fabric binding-enabled switch by issuing the **show fabric-binding status** command.

```
switch# show fabric-binding status
VSAN 1:Activated database
VSAN 4:No Active database
```

## Configuring Switch WWN List

A user-specified fabric binding list contains a list of switch WWNs (sWWNs) within a fabric. If an sWWN attempts to join the fabric, and that sWWN is not on the list or the sWWN is using a domain ID that differs from the one specified in the allowed list, the ISL between the switch and the fabric is automatically isolated in that VSAN and the switch is denied entry into the fabric.

The persistent domain ID can be specified along with the sWWN. Domain ID authorization is required in FICON VSANs where the domains are statically configured and the end devices reject a domain ID change in all switches in the fabric. Domain ID authorization is not required in Fibre Channel VSANs.

To configure a list of sWWNs and domain IDs for a FICON VSAN, follow these steps:

| | Command | Purpose |
|---|---|---|
| Step 1 | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| Step 2 | `switch(config)# fabric-binding database vsan 5`<br>`switch(config-fabric-binding)#` | Enters the fabric binding submode for the specified VSAN. |
| | `switch(config)# no fabric-binding database vsan 5` | Deletes the fabric binding database for the specified VSAN. |
| Step 3 | `switch(config-fabric-binding)# swwn 21:00:05:30:23:11:11:11 domain 102` | Adds the sWWN and domain ID of a switch to the configured database list. |
| | `switch(config-fabric-binding)# swwn 21:00:05:30:23:1a:11:03 domain 101` | Adds the sWWN and domain ID of another switch to the configured database list. |
| | `switch(config-fabric-binding)# no swwn 21:00:15:30:23:1a:11:03 domain 101` | Deletes the sWWN and domain ID of a switch from the configured database list. |
| Step 4 | `switch(config-fabric-binding)# exit`<br>`switch(config)#` | Exits the fabric binding submode. |

To configure a list of sWWNs and optional domain IDs for a Fibre Channel VSAN, follow these steps:

| | Command | Purpose |
|---|---|---|
| Step 1 | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| Step 2 | `switch(config)# fabric-binding database vsan 10`<br>`switch(config-fabric-binding)#` | Enters the fabric binding submode for the specified VSAN. |
| | `switch(config)# no fabric-binding database vsan 10` | Deletes the fabric binding database for the specified VSAN. |
| Step 3 | `switch(config-fabric-binding)# swwn 21:00:05:30:23:11:11:11` | Adds the sWWN of a switch for all domains to the configured database list. |
| | `switch(config-fabric-binding)# no swwn 21:00:05:30:23:11:11:11` | Deletes the sWWN of a switch for all domains from the configured database list. |
| | `switch(config-fabric-binding)# swwn 21:00:05:30:23:1a:11:03 domain 101` | Adds the sWWN of another switch for a specific domain ID to the configured database list. |
| | `switch(config-fabric-binding)# no swwn 21:00:15:30:23:1a:11:03 domain 101` | Deletes the sWWN and domain ID of a switch from the configured database list. |
| Step 4 | `switch(config-fabric-binding)# exit`<br>`switch(config)#` | Exits the fabric binding submode. |

# Fabric Binding Activation

The fabric binding feature maintains a configuration database (config-database) and an active database. The config-database is a read-write database that collects the configurations you perform. These configurations are only enforced upon activation. This activation overwrites the active database with the contents of the config- database. The active database is read-only and is the database that checks each switch that attempts to log in.

By default, the fabric binding feature is not activated. You cannot activate the fabric binding database on the switch if entries existing in the configured database conflict with the current state of the fabric. For example, one of the already logged in switches may be denied login by the config-database. You can choose to forcefully override these situations.

**Note** After activation, any already logged in switch that violates the current active database will be logged out, and all switches that were previously denied login because of fabric binding restrictions are reinitialized.

To activate the fabric binding feature, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# fabric-binding activate vsan 10` | Activates the fabric binding database for the specified VSAN. |
| | `switch(config)# no fabric-binding activate vsan 10` | Deactivates the fabric binding database for the specified VSAN. |

## Forcing Fabric Binding Activation

If the database activation is rejected due to one or more conflicts listed in the previous section, you may decide to proceed with the activation by using the **force** option.

To forcefully activate the fabric binding database, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | `switch(config)# fabric-binding activate vsan 3 force` | Activates the fabric binding database for the specified VSAN forcefully—even if the configuration is not acceptable. |
| | `switch(config)# no fabric-binding activate vsan 3 force` | Reverts to the previously configured state or to the factory default (if no state is configured). |

# Saving Fabric Binding Configurations

When you save the fabric binding configuration, the config database is saved to the running configuration.

**Caution** You cannot disable fabric binding in a FICON-enabled VSAN.

- Use the **fabric-binding database copy vsan** command to copy from the active database to the config database. If the configured database is empty, this command is not accepted.

  ```
  switch# fabric-binding database copy vsan 1
  ```

- Use the **fabric-binding database diff active vsan** command to view the differences between the active database and the config database. This command can be used when resolving conflicts.

  ```
  switch# fabric-binding database diff active vsan 1
  ```

- Use the **fabric-binding database diff config vsan** command to obtain information on the differences between the config database and the active database.

  ```
  switch# fabric-binding database diff config vsan 1
  ```

- Use the **copy running-config startup-config** command to save the running configuration to the startup configuration so that the fabric binding config database is available after a reboot.

  ```
  switch# copy running-config startup-config
  ```

# Clearing the Fabric Binding Statistics

Use the **clear fabric-binding statistics** command to clear all existing statistics from the fabric binding database for a specified VSAN.

```
switch# clear fabric-binding statistics vsan 1
```

# Deleting the Fabric Binding Database

Use the **no fabric-binding** command in configuration mode to delete the configured database for a specified VSAN.

```
switch(config)# no fabric-binding database vsan 10
```

# Verifying Fabric Binding Configurations

Use the **show** commands to display all fabric binding information configured on this switch (see Examples 11-1 to 11-9).

***Example 11-1 Displays Configured Fabric Binding Database Information***

```
switch# show fabric-binding database
-------------------------------------------------
Vsan   Logging-in Switch WWN     Domain-id
-------------------------------------------------
1      21:00:05:30:23:11:11:11   0x66(102)
1      21:00:05:30:23:1a:11:03    0x19(25)
1      20:00:00:05:30:00:2a:1e   0xea(234) [Local]
4      21:00:05:30:23:11:11:11         Any
4      21:00:05:30:23:1a:11:03         Any
4      20:00:00:05:30:00:2a:1e   0xea(234) [Local]
61     21:00:05:30:23:1a:11:03    0x19(25)
61     21:00:05:30:23:11:11:11   0x66(102)
61     20:00:00:05:30:00:2a:1e   0xea(234) [Local]
[Total 7 entries]
```

*Example 11-2 Displays Active Fabric Binding Information*

```
switch# show fabric-binding database active
-------------------------------------------------
Vsan   Logging-in Switch WWN     Domain-id
-------------------------------------------------
1      21:00:05:30:23:11:11:11   0x66(102)
1      21:00:05:30:23:1a:11:03    0x19(25)
1      20:00:00:05:30:00:2a:1e   0xea(234) [Local]
61     21:00:05:30:23:1a:11:03    0x19(25)
61     21:00:05:30:23:11:11:11   0x66(102)
61     20:00:00:05:30:00:2a:1e   0xef(239) [Local]
```

*Example 11-3 Displays Configured VSAN-Specific Fabric Binding Information*

```
switch# show fabric-binding database vsan 4
-------------------------------------------------
Vsan   Logging-in Switch WWN     Domain-id
-------------------------------------------------
4      21:00:05:30:23:11:11:11          Any
4      21:00:05:30:23:1a:11:03          Any
4      20:00:00:05:30:00:2a:1e   0xea(234) [Local]
[Total 2 entries]
```

*Example 11-4 Displays Active VSAN-Specific Fabric Binding Information*

```
switch# show fabric-binding database active vsan 61
-------------------------------------------------
Vsan   Logging-in Switch WWN     Domain-id
-------------------------------------------------
61     21:00:05:30:23:1a:11:03    0x19(25)
61     21:00:05:30:23:11:11:11   0x66(102)
61     20:00:00:05:30:00:2a:1e   0xef(239) [Local]
[Total 3 entries]
```

*Example 11-5 Displays Fabric Binding Statistics*

```
switch# show fabric-binding statistics
Statistics For VSAN: 1
-----------------------
Number of sWWN permit: 0
Number of sWWN deny  : 0

Total Logins permitted  : 0
Total Logins denied     : 0
Statistics For VSAN: 4
-----------------------
Number of sWWN permit: 0
Number of sWWN deny  : 0

Total Logins permitted  : 0
Total Logins denied     : 0
Statistics For VSAN: 61
-----------------------
Number of sWWN permit: 0
Number of sWWN deny  : 0

Total Logins permitted  : 0
Total Logins denied     : 0
```

```
Statistics For VSAN: 345
------------------------
Number of sWWN permit: 0
Number of sWWN deny  : 0

Total Logins permitted  : 0
Total Logins denied     : 0
Statistics For VSAN: 346
------------------------
Number of sWWN permit: 0
Number of sWWN deny  : 0

Total Logins permitted  : 0
Total Logins denied     : 0
Statistics For VSAN: 347
------------------------
Number of sWWN permit: 0
Number of sWWN deny  : 0

Total Logins permitted  : 0
Total Logins denied     : 0
Statistics For VSAN: 348
------------------------
Number of sWWN permit: 0
Number of sWWN deny  : 0

Total Logins permitted  : 0
Total Logins denied     : 0
Statistics For VSAN: 789
------------------------
Number of sWWN permit: 0
Number of sWWN deny  : 0

Total Logins permitted  : 0
Total Logins denied     : 0
Statistics For VSAN: 790
------------------------
Number of sWWN permit: 0
Number of sWWN deny  : 0

Total Logins permitted  : 0
Total Logins denied     : 0
```

***Example 11-6 Displays Fabric Binding Status for Each VSAN***

```
switch# show fabric-binding status
VSAN 1 :Activated database
VSAN 4 :No Active database
VSAN 61 :Activated database
VSAN 345 :No Active database
VSAN 346 :No Active database
VSAN 347 :No Active database
VSAN 348 :No Active database
VSAN 789 :No Active database
VSAN 790 :No Active database
```

***Example 11-7 Displays Fabric Binding Violations***

```
switch# show fabric-binding violations
--------------------------------------------------------------------------------
```

```
VSAN Switch WWN [domain]     Last-Time              [Repeat count] Reason
-------------------------------------------------------------------------------
2    20:00:00:05:30:00:4a:1e [0xeb] Nov 25 05:46:14 2003   [2]    Domain mismatch
3    20:00:00:05:30:00:4a:1e [*] Nov 25 05:44:58 2003      [2]    sWWN not found
4    20:00:00:05:30:00:4a:1e [*] Nov 25 05:46:25 2003      [1]    Database mismatch
```

**Note** In VSAN 3 the sWWN itself was not found in the list. In VSAN 2, the sWWN was found in the list, but has a domain ID mismatch.

***Example 11-8 Displays EFMD Statistics***

```
switch# show fabric-binding efmd statistics

EFMD Protocol Statistics for VSAN 1
---------------------------------------
Merge Requests -> Transmitted : 0 , Received : 0
Merge Accepts  -> Transmitted : 0 , Received : 0
Merge Rejects  -> Transmitted : 0 , Received : 0
Merge Busy     -> Transmitted : 0 , Received : 0
Merge Errors   -> Transmitted : 0 , Received : 0

EFMD Protocol Statistics for VSAN 4
---------------------------------------
Merge Requests -> Transmitted : 0 , Received : 0
Merge Accepts  -> Transmitted : 0 , Received : 0
Merge Rejects  -> Transmitted : 0 , Received : 0
Merge Busy     -> Transmitted : 0 , Received : 0
Merge Errors   -> Transmitted : 0 , Received : 0

EFMD Protocol Statistics for VSAN 61
---------------------------------------
Merge Requests -> Transmitted : 0 , Received : 0
Merge Accepts  -> Transmitted : 0 , Received : 0
Merge Rejects  -> Transmitted : 0 , Received : 0
Merge Busy     -> Transmitted : 0 , Received : 0
Merge Errors   -> Transmitted : 0 , Received : 0
```

***Example 11-9 Displays EFMD Statistics for a Specified VSAN***

```
switch# show fabric-binding efmd statistics vsan 4

EFMD Protocol Statistics for VSAN 4
---------------------------------------
Merge Requests -> Transmitted : 0 , Received : 0
Merge Accepts  -> Transmitted : 0 , Received : 0
Merge Rejects  -> Transmitted : 0 , Received : 0
Merge Busy     -> Transmitted : 0 , Received : 0
Merge Errors   -> Transmitted : 0 , Received : 0
```

# Default Settings

Table 11-2 lists the default settings for the fabric binding feature.

*Table 11-2 Default Fabric Binding Settings*

| Parameters | Default |
|---|---|
| Fabric binding | Disabled |

CHAPTER 12

# Configuring Cisco TrustSec Fibre Channel Link Encryption

This chapter provides an overview of the Cisco TrustSec Fibre Channel (FC) Link Encryption feature and describes how to configure and set up link-level encryption between switches.

The chapter includes the following sections:



## Cisco TrustSec FC Link Encryption Terminology

The following Cisco TrustSec FC Link Encryption-related terms are used in this chapter:

- Galois Counter Mode (GCM)—A block cipher mode of operation providing confidentiality and data-origin authentication.
- Galois Message Authentication Code (GMAC)—A block cipher mode of operation providing only data-origin authentication. It is the authentication-only variant of GCM.
- Security Association (SA)—A connection that handles the security credentials and controls how they propagate between switches. The SA includes parameters such as salt and keys.
- Key—A 128-bit hexadecimal string that is used for frame encryption and decryption. The default value is zero.
- Salt —A 32-bit hexadecimal number that is used during encryption and decryption. The same salt must be configured on both sides of the connection to ensure proper communication. The default value is zero.
- Security Parameters Index (SPI) number—A 32-bit number that identifies the SA to be configured to the hardware. The range is from 256 to 65536.

# Support for AES Encryption

The Advanced Encryption Standard (AES) is the symmetric cipher algorithm that provides a high-level of security, and can accept different key sizes.

The Cisco TrustSec FC Link Encryption feature supports the 128-bit AES for security encryption and enables either AES-GCM or AES-GMAC for an interface. The AES-GCM mode provides encryption and authentication of the frames and AES-GMAC provides only the authentication of the frames that are being passed between the two peers.

# About Cisco TrustSec FC Link Encryption

Cisco TrustSec FC Link Encryption is an extension of the Fibre Channel-Security Protocol (FC-SP) feature and uses the existing FC-SP architecture to provide integrity and confidentiality of transactions. Encryption is now added to the peer authentication capability to provide security and prevent unwanted traffic interception. Peer authentication is implemented according to the FC-SP standard using the Diffie-Hellman Challenge Handshake Authentication Protocol (DHCHAP) protocol.

**Note** Cisco TrustSec FC Link Encryption is currently only supported between Cisco MDS switches. This feature is not supported when you downgrade to software versions which do not have the Encapsulating Security Protocol (ESP) support.

This section includes the following topics:



## Supported Modules

The following modules are supported for the Cisco TrustSec FC Link Encryption feature:

- 2/4/8/10/16 Gbps 48-ports Advanced Fibre Channel module (DS-X9448-768K9)
- 32-port 8-Gbps Advanced Fibre Channel Switching module (DS-X9232-256K9)
- 48-port 8-Gbps Advanced Fibre Channel Switching module (DS-X9248-256K9)
- 1/2/4/8 Gbps 24-Port Fibre Channel switching module (DS-X9224-96K9)
- 1/2/4/8 Gbps 48-Port Fibre Channel switching module (DS-X9248-96K9)
- 1/2/4/8 Gbps 4/44-Port Fibre Channel switching module (DS-X9248-48K9)
- 2/4/8/10/16 Gbps 96-ports Fibre Channel Switching Module (DS-C9396S-K9)
- 24/10 port SAN Extension module (DS-X9334-K9)

**Note** 24/10 port SAN Extension module (DS-X9334-K9) is supported on Cisco MDS 9700 Series Directors starting from Cisco MDS NX-OS Release 7.3(0)DY(1).

# Enabling Cisco TrustSec FC Link Encryption

By default, the FC-SP feature and the Cisco TrustSec FC Link Encryption feature are disabled in all switches in the Cisco MDS 9000 Family.

You must explicitly enable the FC-SP feature to access the configuration and verification commands for fabric authentication and encryption. When you disable this feature, all related configurations are automatically discarded.

To enable FC-SP for a Cisco MDS switch, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# feature fcsp` | Enables the FC-SP feature. |
| | `switch(config)# no feature fcsp` | Disables (default) the FC-SP feature in this switch. |

Configuring the Cisco TrustSec FC Link Encryption feature requires the ENTERPRISE_PKG license. For more information, refer to the *Cisco MDS 9000 Family NX-OS Licensing Guide*.

# Setting Up Security Associations

To perform encryption between the switches, a security association (SA) needs to be set up. An administrator manually configures the SA before the encryption can take place. The SA includes parameters such as keys and salt, that are required for encryption. You can set up to 2000 SAs in a switch.

To set up an SA between two switches, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# fcsp esp sa` *spi_number* | Enters into SA submode for configuring SAs. The range of *spi_number* is from 256 to 65536. |
| **Step 3** | `switch(config)# no fcsp esp sa` *spi_number* | Deletes the SA between the switches.[1] |

1. If the specified SA is currently programmed to the ports, this command returns an error saying that the SA is in use.

To determine which ports are using the SA, use the **show running-config fcsp** command. Refer to the "Viewing Running System Information" section on page 12-265.

**Note** Cisco TrustSec FC Link Encryption is currently supported only on DHCHAP on and off modes.

# Setting Up Security Association Parameters

To set up the SA parameters, such as keys and salt, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters configuration mode. |
| **Step 2** | `switch(config)# fcsp esp sa spi_number` | Enters into SA submode for configuring SAs. The range of *spi_number* is from 256 to 65536. |
| **Step 3** | | |
| **Step 4** | `switch(config-sa)# key key` | Configures the key for the SA. Maximum size of *key* is 34. |
| **Step 5** | `switch(config-sa)# no key key` | Removes the key from the SA. |
| **Step 6** | `switch(config-sa)# salt salt` | Configures the salt for the SA. The range is from 0x0 to 0xffffffff. |
| **Step 7** | `switch(config-sa)# no salt salt` | Removes the salt for the SA. |

# Configuring ESP Settings

This section includes the following topics:



## Configuring ESP on Ingress and Egress Ports

Once the SA is created, you need to configure Encapsulating Security Protocol (ESP) on the ports. You should specify the egress and ingress ports for the encryption and decryption of packets between the network peers. The egress SA specifies which keys or parameters are to be used for encrypting the packets that leave the switch. The ingress SA specifies which keys or parameters are to be used to decrypt the packets entering that particular port.

This section covers the following topics:



### Configuring ESP on Ingress Port

To configure SA to the ingress hardware, follow these steps:

| | | |
|---|---|---|
| **Step 1** | `switch# config t` | Enters the configuration mode. |
| **Step 2** | `switch(config)# interface fc x/y` | Configures the FC interface on slot *x*, port *y*.<br>**Note** Selecting a portchannel will apply the configuration on all members of the portchannel. |
| **Step 3** | `switch(config-if)# fcsp esp manual` | Enters the ESP configuration submode. |
| **Step 4** | `switch(config-if-esp)# ingress-sa spi_number` | Configures the SA to the ingress hardware. |
| **Step 5** | `switch (config-if-esp)# no ingress-sa spi_number` | Removes the SA from the ingress hardware.[1] |

1. If SA is not configured in the ingress port, then running this command returns an error message.

### Configuring ESP on Egress Ports

To configure SA to the egress hardware, follow these steps:

| | | |
|---|---|---|
| **Step 1** | `switch# config t` | Enters the configuration mode. |
| **Step 2** | `switch(config)# interface fc x/y` | Configures the FC interface on slot *x*, port *y*.<br>**Note** Selecting a portchannel will apply the configuration on all members of the portchannel. |
| **Step 3** | `switch(config-if)# fcsp esp manual` | Enters the ESP configuration submode. |
| **Step 4** | `switch(config-if-esp)# egress-sa spi_number` | Configures the SA to the egress hardware. |
| **Step 5** | `switch(config-if)# no fcsp esp manual` | Removes the SA from the ingress and egress hardware.[1] |

1. If SA is not configured in the egress port, then running this command returns an error message.

**Note** To apply the SA to the ingress and egress hardware of an interface, the interface needs to be in the admin shut mode.

## Configuring ESP Modes

Configure the ESP settings for the ports as GCM to enable message authentication and encryption or as GMAC to enable message authentication.

The default ESP mode is AES-GCM.

This section covers the following topics:

- Configuring AES-GCM, page 12-263
- Configuring AES-GMAC, page 12-264

### Configuring AES-GCM

To configure the AES-GCM mode, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters the configuration mode. |
| **Step 2**<br>**Step 3**<br>**Step 4** | `switch(config)# interface fc x/y` | Configures the FC interface on slot *x*, port *y*.<br>**Note** Selecting a portchannel would apply the configuration on all members of the portchannel. |
| | `switch(config-if)# fcsp esp manual` | Enters the ESP configuration submode to configure the ESP settings on each port. |
| **Step 5** | `switch(config-if-esp)# mode gcm` | Sets the GCM mode for the interface. |

### Configuring AES-GMAC

To configure AES-GMAC mode, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `switch# config t` | Enters the configuration mode. |
| **Step 2** | `switch(config)# interface fc x/y` | Configures the FC interface on slot *x*, port *y*. |
| **Step 3** | | **Note** Selecting a portchannel would apply the configuration on all members of the portchannel. |
| **Step 4** | `switch(config-if)# fcsp esp manual` | Enters the ESP configuration submode to configure the ESP settings on each port. |
| **Step 5** | `switch(config-if-esp)# mode gmac` | Sets the GMAC mode for the interface. |
| **Step 6** | `switch(config-if-esp)# no mode gmac` | Removes the GMAC mode from the interface and applies the default AES-GCM mode. |

**Note** The ESP modes are set only after a SA is configured to either the ingress or the egress hardware. If SA has not been configured, ESP is turned off and encapsulation does not occur.

**Note** An ESP mode change always needs a port flap because the change is not seamless if it is done after you configure the port; although the configurations are not rejected.

**Note** Only ISLs with FC-SP port mode turned on and available on ESP capable switches or blades are displayed.

**Note** You can modify an existing ESP configuration provided the selected ISLs are enabled.

# Viewing Cisco TrustSec FC Link Encryption Information

You can view information about the Cisco TrustSec FC Link Encryption feature using the **show** commands Fabric Manager or Device Manager.

This section covers the following topics:



# Viewing FC-SP Interface Information

Use the **show fcsp interface** command to show all FC-SP-related information for a specific interface.

```
switch# show fcsp interface fc7/41
```

```
fc7/41:
        fcsp authentication mode:SEC_MODE_OFF
        ESP is enabled
        configured mode is: GCM
        programmed ingress SA: 300, 303
        programmed egress SA: 300
        Status:FC-SP protocol in progress
```

# Viewing Running System Information

Use the **show running-config fcsp** command to show all the run-time information relevant to FC-SP. All details about ESP and configured interfaces are displayed. Use this command to determine which ports are using SA.

```
switch# show running-config fcsp
version 4.1(2)
feature fcsp
fcsp esp sa 300
  key 0x00000000000000000000000000123456
  salt 0x123456
fcsp esp sa 301
  key 0x00000000000000000000000000123456
  salt 0x1234567
fcsp esp sa 302
  key 0x00000000000000000000000000123456
  salt 0x123456

interface fc8/48
  fcsp off
  fcsp esp manual
    ingress-sa 300
    ingress-sa 301
    egress-sa 300
```

# Viewing FC-SP Interface Statistics

Use the **show fcsp interface statistics** command to show all statistics related to DHCHAP and ESP for an interface. The ESP statistics shown depend on the ESP supported by the port ASIC.

```
switch# show fcsp interface fc3/31 statistics

fc7/41:
       fcsp authentication mode:SEC_MODE_ON
        ESP is enabled
        configured mode is: GMAC
        programmed ingress SA: 256, 257
        programmed egress SA: 256
        Status:Successfully authenticated
       Authenticated using local password database
        Statistics:
        FC-SP Authentication Succeeded:17
        FC-SP Authentication Failed:3
        FC-SP Authentication Bypassed:0
        FC-SP ESP SPI Mismatched frames:0
        FC-SP ESP Auth failed frames:0
```

# Cisco TrustSec FC Link Encryption Best Practices

Best practices are the recommended steps that should be taken to ensure the proper operation of Cisco TrustSec FC Link Encryption.

This section covers the following topics:



## General Best Practices

This section lists the general best practices for Cisco TrustSec FC Link Encryption:

- Ensure that Cisco TrustSec FC Link Encryption is enabled only between MDS switches. This feature is supported only on E-ports or the ISLs, and errors will result if non-MDS switches are used.
- Ensure that the peers in the connection have the same configurations. If there are differences in the configurations, a “port re-init limit exceeded” error message is displayed.
- Before applying the SA to the ingress and egress hardware of a switch interface, ensure that the interface is in the admin shut mode.

## Best Practices for Changing Keys

After the SA is applied to the ingress and egress ports, you should change the keys periodically in the configuration. The keys should be changed sequentially to avoid traffic disruption.

As an example, consider that a security association has been created between two switches, Switch1 and Switch2. The SA is configured on the ingress and egress ports as shown in the following example:

```
switch# config t
switch(config)# interface fc1/1
switch(config-if)# fcsp esp manual
switch(config-if)# ingress-sa 256
switch(config-if)# egress-sa 256
```

To change the keys for these switches, follow these steps:

**Step 1** Add a new SA on Switch1 and Switch2.

```
switch# config t
switch(config)# fcsp esp sa 257
switch(config-sa)# key 0xAC9EF8BC8DB2DBD2008D184F794E0C38
switch(config-sa)# salt 0x1234
```

**Step 2** Configure the ingress SA on Switch1.

```
switch# config t
switch(config)# interface fc1/1
switch(config-if)# fcsp esp manual
switch(config-if)# ingress-sa 257
```

**Step 3** Configure the ingress and the egress SA on Switch2.

```
switch# config t
switch(config)# interface fc1/1
switch(config-if)# fcsp esp manual
```

```
switch(config-if)# ingress-sa 257
switch(config-if)# egress-sa 257
```

**Step 4** Configure the egress SA on Switch1.

```
switch# config t
switch(config)# interface fc1/1
switch(config-if)# fcsp esp manual
switch(config-if)# egress-sa 257
```

**Step 5** Remove the previously configured ingress SA from both the switches.

```
switch# config t
switch(config)# interface fc1/1
switch(config-if)# fcsp esp manual
switch(config-if)# no ingress-sa 256
```