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` `switch(config-role)#`	Places you in the mode for the specified role (techdocs). **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` `switch(config-role)#`	Places you in role configuration submode for the existing role sangroup.

Step 3	`switch(config-role)# rule 1 permit config`
	`switch(config-role)# rule 2 deny config feature fspf`
	`switch(config-role)# rule 3 permit debug feature zone`
	`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` `switch(config-role)#`	Places you in role configuration submode for the sangroup role.
Step 3	`switch(config)# vsan policy deny` `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` `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` `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` `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.

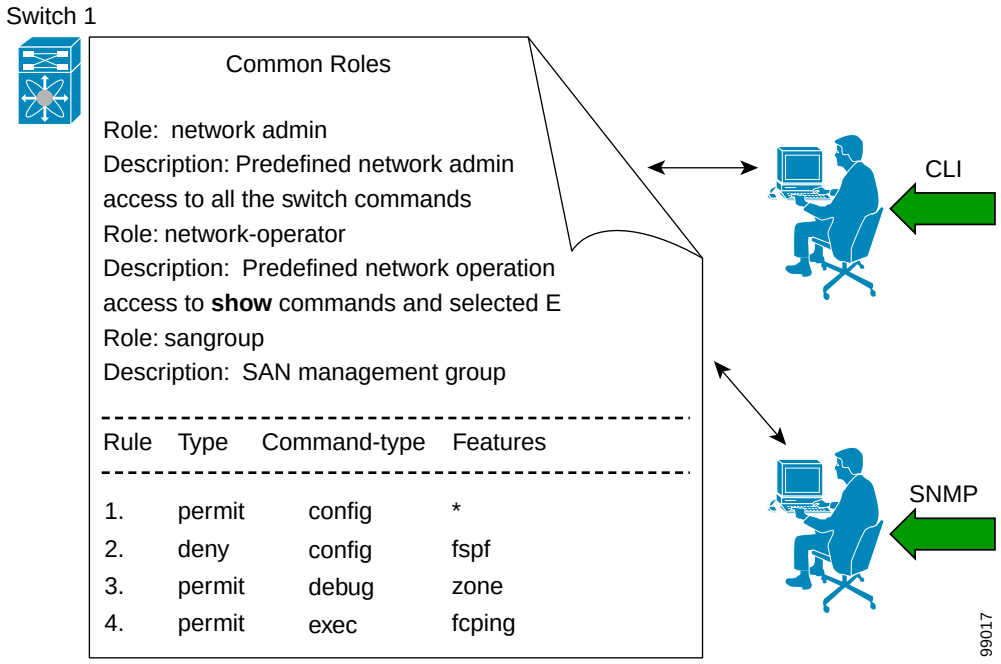
Example 3-4 Displays the Differences Between the Two Databases

```
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). **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`
	`generating dsa key.....`
	`generated dsa key`
		Generates the DSA server key pair.

	`switch(config)# ssh key rsa 1024`
	`generating rsa key.....`
	`generated rsa key`
		Generates the RSA server key pair.

	`switch(config)# no ssh key rsa 1024` `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` `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` `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` `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`
	`ssh key dsa 512`
	`dsa keys already present, use force`
	`option to overwrite them`
	`switch(config)# ssh key dsa 512 force`
	`deleting old dsa key.....`
	`generating dsa key.....`
	`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.
		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` `updated`	Enables the use of the SSH service.
	`switch(config)# no feature ssh` `updated`	Disables (default) the use of the SSH service.
	`switch(config)# feature telnet` `updated`	Enables the use of the Telnet service.
	`switch(config)# no feature telnet` `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`
	`generating rsa key(1024 bits).....`
	`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.
		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`
	`**********`
	`rsa Keys generated: Thu Jul 9 11:10:29 2009`
	`ssh-rsa`
	`AAAAB3NzaC1yc2EAAAABIwAAAIEAxWmjJT+oQhIcvnrMbx2BmD`
	`0P8boZElTfJFx9fexWp6rOiztlwODtehnjadWc6A+DE2DvYNvq`
	`srU9TBypYDPQkR/+Y6cKubyFWVxSBG/NHztQc3+QC1zdkIxGNJ`
	`bEHyFoajzNEO8LLOVFIMCZ2Td7gxUGRZc+fbqS33GZsCAX6v0=`
	`bitcount:262144`
	`fingerprint:`
	`8d:44:ee:6c:ca:0b:44:95:36:d0:7d:f2:b5:78:74:7d`
	`**********`
	`could not retrieve dsa key information`
	`**********`
		Shows the public key for the account (admin).

Step 4	`switch(config)# username admin keypair export bootflash:key_rsa rsa`
	`Enter Passphrase:`
	`switch(config)# dir`
	`951 Jul 09 11:13:59 2009 key_rsa`
	`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.
		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.
		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`
	`Enter Passphrase:`
	`switch(config)# show username admin keypair`
	`**********`
	`rsa Keys generated: Thu Jul 9 11:10:29 2009`
	`ssh-rsa`
	`AAAAB3NzaC1yc2EAAAABIwAAAIEAxWmjJT+oQhIcvnrMbx2BmD`
	`0P8boZElTfJFx9fexWp6rOiztlwODtehnjadWc6A+DE2DvYNvq`
	`srU9TBypYDPQkR/+Y6cKubyFWVxSBG/NHztQc3+QC1zdkIxGNJ`
	`bEHyFoajzNEO8LLOVFIMCZ2Td7gxUGRZc+fbqS33GZsCAX6v0=`
	`bitcount:262144`
	`fingerprint:`
	`8d:44:ee:6c:ca:0b:44:95:36:d0:7d:f2:b5:78:74:7d`
	`**********`
	`could not retrieve dsa key information`
	`**********`
		Imports the keypair to the home directory of the switch.
		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.
		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.
		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