BROCADE

CONFIGURATION GUIDE

Network OS Security Configuration Guide, v7.0.1

Supporting Network OS 7.0.1

53-1004364-01
20 May 2016

Contents

Preface

Document conventions

The document conventions describe text formatting conventions, command syntax conventions, and important notice formats used in Brocade technical documentation.

Text formatting conventions

Text formatting conventions such as boldface, italic, or Courier font may be used in the flow of the text to highlight specific words or phrases.

Format	Description
bold text	Identifies command names
	Identifies keywords and operands
	Identifies the names of user-manipulated GUI elements
	Identifies text to enter at the GUI
italic text	Identifies emphasis
	Identifies variables
	Identifies document titles
`Courier font`	Identifies CLI output
	Identifies command syntax examples

Command syntax conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

Convention	Description
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.
value	In Fibre Channel products, a fixed value provided as input to a command option is printed in plain text, for example, **--show** WWN.
[ ]	Syntax components displayed within square brackets are optional.
	Default responses to system prompts are enclosed in square brackets.
{ **x** | **y** | **z** }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options.
	In Fibre Channel products, square brackets may be used instead for this purpose.

Convention	Description
x | y	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, *member*[*member*...].
\	Indicates a "soft" line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Notes, cautions, and warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential hazards.

NOTE

A Note provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

ATTENTION

An Attention statement indicates a stronger note, for example, to alert you when traffic might be interrupted or the device might reboot.

CAUTION

A Caution statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.

DANGER

A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Brocade resources

Visit the Brocade website to locate related documentation for your product and additional Brocade resources.

You can download additional publications supporting your product at www.brocade.com. Select the Brocade Products tab to locate your product, then click the Brocade product name or image to open the individual product page. The user manuals are available in the resources module at the bottom of the page under the Documentation category.

To get up-to-the-minute information on Brocade products and resources, go to MyBrocade. You can register at no cost to obtain a user ID and password.

Release notes are available on MyBrocade under Product Downloads.

White papers, online demonstrations, and data sheets are available through the Brocade website.

Contacting Brocade Technical Support

As a Brocade customer, you can contact Brocade Technical Support 24x7 online, by telephone, or by e-mail. Brocade OEM customers contact their OEM/Solutions provider.

Brocade customers

For product support information and the latest information on contacting the Technical Assistance Center, go to http://www.brocade.com/services-support/index.html.

If you have purchased Brocade product support directly from Brocade, use one of the following methods to contact the Brocade Technical Assistance Center 24x7.

Online	Telephone	E-mail
Preferred method of contact for non-urgent issues: • My Cases through MyBrocade • Software downloads and licensing tools • Knowledge Base	Required for Sev 1-Critical and Sev 2-High issues: • Continental US: 1-800-752-8061 • Europe, Middle East, Africa, and Asia Pacific: +800-AT FIBREE (+800 28 34 27 33) • For areas unable to access toll free number: +1-408-333-6061 • Toll-free numbers are available in many countries.	support@brocade.com Please include: • Problem summary • Serial number • Installation details • Environment description

Brocade OEM customers

If you have purchased Brocade product support from a Brocade OEM/Solution Provider, contact your OEM/Solution Provider for all of your product support needs.

- OEM/Solution Providers are trained and certified by Brocade to support Brocade® products.
- Brocade provides backline support for issues that cannot be resolved by the OEM/Solution Provider.
- Brocade Supplemental Support augments your existing OEM support contract, providing direct access to Brocade expertise. For more information, contact Brocade or your OEM.
- For questions regarding service levels and response times, contact your OEM/Solution Provider.

Document feedback

To send feedback and report errors in the documentation you can use the feedback form posted with the document or you can e-mail the documentation team.

Quality is our first concern at Brocade and we have made every effort to ensure the accuracy and completeness of this document. However, if you find an error or an omission, or you think that a topic needs further development, we want to hear from you. You can provide feedback in two ways:

- Through the online feedback form in the HTML documents posted on www.brocade.com.
- By sending your feedback to documentation@brocade.com.

Provide the publication title, part number, and as much detail as possible, including the topic heading and page number if applicable, as well as your suggestions for improvement.

About this document

Supported hardware and software

In those instances in which procedures or parts of procedures documented here apply to some devices but not to others, this guide identifies exactly which devices are supported and which are not.

Although many different software and hardware configurations are tested and supported by Brocade Communications Systems, Inc. for Network OS, documenting all possible configurations and scenarios is beyond the scope of this document.

The following hardware platforms are supported by this release of Network OS:

- Brocade VDX 2741
- Brocade VDX 2746
- Brocade VDX 6740
  - Brocade VDX 6740-48
  - Brocade VDX 6740-64
- Brocade VDX 6740T
  - Brocade VDX 6740T-48
  - Brocade VDX 6740T-64
  - Brocade VDX 6740T-1G
- Brocade VDX 6940-36Q
- Brocade VDX 6940-144S
- Brocade VDX 8770
  - Brocade VDX 8770-4
  - Brocade VDX 8770-8

To obtain information about a Network OS version other than this release, refer to the documentation specific to that version.

Using the Network OS CLI

For complete instructions and support for using the Network OS command line interface (CLI), refer to the *Network OS Command Reference*.

What's new in this document

This document describes the concepts and configuration of the security features for Network OS.

The content has been updated with the following changes for Network OS v7.0.1 :

- No significant changes

The content has been updated with the following changes for Network OS v7.0.0 :

- Updates for Certificate Management for HTTPS
- Updates for SSH encryption protocol
- Updates for ACL configuration support
- Updates forTACACS+ configuration support

Managing User Accounts

Understanding and managing user accounts

A user account allows authorized user access to the device Command Line Interface (CLI). A user account must be assigned a role to specify the account's access privileges. A user account can be disabled at any point, preventing the user from logging in to the device. A user can only be unlocked when the account is auto-locked because the user exceeded the configured threshold for failed login attempts. Only an administrator can create, change, unlock, or delete user accounts.

All modules that pertain to security, for example, user and user roles, role-based access control (RBAC), and password attributes (for example, encryption), are globally configurable data entities.

This means that if a Network OS device is in logical chassis cluster mode, all Network OS devices in the cluster have a common configuration for all the mentioned entities.

Default accounts in the local switch user database

The device software comes with two predefined user accounts that are part of the factory-default settings. Brocade recommends that you change the password for all default accounts during the initial installation and configuration for each device.

The default user accounts are "admin" and "user," and these accounts are associated with the corresponding admin" and "user" roles in the device-local user database. Only the "admin" and "user" users can access the CLI and, except for the account password, no other attributes can be changed for the default users "admin" and "user."

By default, all account information is stored in the device-local user database. User authentication and tracking of logins to the device is local by default.

NOTE

The maximum number of user accounts, including the default accounts, is 64. The maximum number of roles, including the default roles is 64. For any environment requiring more than 64 users, you should adopt an authentication, authorization, and accounting (AAA) service for user management. Refer to Managing User Accounts on page 13 for more information. The maximum number of active Telnet or CLI sessions supported per switch is 32.

User account attributes

The following table summarizes the available user account attributes.

TABLE 1 User account attributes

Parameter	Description
name	The name of the account. The user account name is case-sensitive, must not exceed 40 characters, and must begin with a letter or an underscore. The text string can contain letters, numbers, underscores (__), and periods (.). If the user name specified already exists, the **username** command modifies the existing role.

TABLE 1 User account attributes (continued)

Parameter	Description
	NOTE If a user name with a leading underscore exists, a firmware downgrade would be blocked, and this user name would need to be removed or changed.
role	The role assigned to the user defines the RBAC access privileges for the account.
password	The account password must satisfy all currently enforced password rules. Refer to Password policies overview on page 17 for more information.
encryption-level	The password encryption level. You can choose to encrypt the password (7) or leave it in clear text (0). If you do not specify an encryption level, encrypted (7) is the default level.
desc	A description of the account. The description can be up to 64 characters long, and can include any printable ASCII character, except for the following characters: single quotation marks (‘), double quotation marks(“), exclamation point (!), colon (:), and semi-colon (;). If the description contains spaces. you must enclose the text in double quotation marks.
enable true | false	Indicates whether the account is enabled or disabled. A user whose account is disabled cannot log in. The default account status is enabled.

Configuring user accounts

When you create a user account you must specify three mandatory attributes: an account login name, a role, and a password. The remaining attributes are optional.

Creating a user account

The following example creates a new user account with the minimally required attributes: name, role, and password. The account name "brcdUser" has the default user privilege of accessing commands in privileged EXEC mode. Refer to the *Network OS Command Reference* for complete options on usernames.

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.
2. Enter the **username** command with the specified parameters.

```
switch# configure terminal
Entering configuration mode terminal
switch(config)# username brcdUser role user password welcome
```

Examples

Use the **show running-config username** command in privileged EXEC mode to display all configured users.

```
switch# show running-config username
username admin password "BwrsDbB+tABWGWpINOVKoQ==\n" encryption-level 7 role admin desc Administrator
username user password "BwrsDbB+tABWGWpINOVKoQ==\n" encryption-level 7 role user desc User
```

Use the **show running-config username** *username* command in privileged EXEC mode to display a single user.

```
switch# show running-config username admin
username admin password "BwrsDbB+tABWGWpINOVKoQ==\n" encryption-level 7 role admin desc Administrator
```

Use the **show running-config username** *username* **enable** command in privileged EXEC mode to display whether the account is enabled or disabled.

```
switch# show running-config username admin enable
username admin enable true
```

Modifying an existing user account

The syntax for the account *create* and *modify* operations is essentially the same. The difference is that there are no mandatory parameters for modifying an existing account. The system internally recognizes whether a new account is created or an existing account is modified by checking whether the user account is already present in the configuration database.

The following example adds a description to the previously created "brcdUser" account.

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.
2. Enter the **username** command with the specified parameters.

```
switch# configure terminal
Entering configuration mode terminal
switch(config)# username brcdUser
switch(config-username-brcdUser)# desc "Brocade guest account"
```

Disabling a user account

You can disable a user account by setting the **enable** parameter to **false**. All active login sessions for a user are terminated when a user account is disabled.

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.
2. Enter the **username** command with the specified parameters.

```
switch# configure terminal
Entering configuration mode terminal
switch(config)# username testUser enable false
```

Deleting a user account

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.
2. Enter the **no username** command.

```
switch# configure terminal
Entering configuration mode terminal
switch(config)# no username testUser
```

All active login sessions for a user are terminated when a user account is deleted.

Unlocking a user account

A user account is automatically locked by the system when the configured threshold for repeated failed login attempts has been reached. The account lockout threshold is a configurable parameter. Refer to Account lockout policy on page 18 for more information.

If a user account is locked out of a switch, that same user can still try to log in on another switch in the cluster. However, the unlocking is done on the given RBridge IDs, irrespective of whether the user is not locked or not on one or more switches.

The following procedure shows the commands used to unlock a user account.

NOTE

The **username** and **no username** commands are global configuration commands, but the **unlock username** command is a privileged EXEC command.

1. Enter the **show users** command in privileged EXEC mode to display currently active sessions and locked out users.

```
switch# show users
**USER SESSIONS**
RBridge
ID  Username    Host Ip       Method       Time Logged In        TTY
2   user        10.70.4.105   cli        2012-04-30 01:59:35    pts/2
1   user        10.70.4.105   cli       2012-04-30 01:57:41    pts/2
1   admin       10.70.4.105   cli        2012-04-30 01:58:41    pts/1
1   user        10.70.4.105   cli        2012-09-30 02:04:42    tty80
**LOCKED USERS**
RBridge
ID   username
1    testUser
```

2. Enter the **unlock username** command in privileged EXEC mode to unlock the locked user account.

```
switch#  unlock username testUser
Result: Unlocking the user account is successful
```

3. Verify that the user account has been unlocked. The **show users** command should display "no locked users".

```
switch#  show users
**USER SESSIONS**
RBridge
ID  Username    Host Ip       Method       Time Logged In        TTY
2   user        10.70.4.105   cli        2012-04-30 01:59:35    pts/2
1   user        10.70.4.105   cli       2012-04-30 01:57:41    pts/2
1   admin       10.70.4.105   cli        2012-04-30 01:58:41    pts/1
1   user        10.70.4.105   cli        2012-09-30 02:04:42    tty80
**LOCKED USERS**
RBridge
ID   username
no locked users
```

Configuring a user alias

You can specify a global alias and user alias for the switch by using the **alias** command. The **alias** command operates in two slightly different ways, depending on which configuration mode you are using; global alias or user-level alias. The global alias is accessible by all users. The user-level alias is accessible only when the respective user logs in.

To set a global alias and a user alias, perform the following task in global configuration mode.

1. Enter alias configuration mode.

```
switch(config)# alias-config
```

2. Set the global user-alias for the switch.

```
switch(config-alias-config)# alias redwood engineering
```

3. Enter user configuration mode.

```
switch(config-alias-config)# user john smith
```

4. Set the user-level alias.

```
switch(config-alias-config-user)# alias manager engineering
```

Understanding and managing password policies

Password policies overview

Password policies define and enforce a set of rules that make passwords more secure by subjecting all new passwords to global restrictions. The password policies described in this section apply to the device-local user database only. Configured password policies (and all user account attributes and password state data) are synchronized across management modules and remain unchanged after an HA failover.

In logical chassis cluster mode, the configuration is applied to all the nodes in the cluster.

The following three subsections detail the configurable password policies.

Password strength policy

The following table lists configurable password policy parameters.

TABLE 2 Password policy parameters

Parameter	Description
character-restriction lower	Specifies the minimum number of lowercase alphabetic characters that must occur in the password. The maximum value must be less than or equal to the minimum length value. The default value is zero, which means there is no restriction of lowercase characters.
character-restriction upper	Specifies the minimum number of uppercase alphabetic characters that must occur in the password. The maximum value must be less than or equal to the Minimum Length value. The default value is zero, which means there is no restriction of uppercase characters.
character-restriction numeric	Specifies the minimum number of numeric characters that must occur in the password. The maximum value must be less than or equal to the Minimum Length value. The default value is zero, which means there is no restriction of numeric characters.
character-restriction special-char	Specifies the minimum number of punctuation characters that must occur in the password. All printable, non-alphanumeric punctuation characters except the colon (:) are allowed. The value must be less than or equal to the Minimum Length value. The default value is zero, which means there is no restriction of punctuation characters. Characters added after an exclamation point are dropped. For example, if you use the password "first!second", the password will become "first!" Special characters, such as backslash (\) and question mark (?), are not counted as characters in a password unless the password is specified within quotes.
min-length	Specifies the minimum length of the password. Passwords must be from 8 through 32 characters in length. The default value is 8. The total of the previous four parameters (lowercase, uppercase, digits, and punctuation) must be less than or equal to the Minimum Length value.
max-retry	Specifies the number of failed password logins permitted before a user is locked out. The lockout threshold can range from 0 through 16. The default value is 0. When a password fails more than one of the strength attributes, an error is reported for only one of the attributes at a time.

NOTE

Passwords can have a maximum of 40 characters.

Password encryption policy

The software supports encrypting the passwords of all existing user accounts by enabling password encryption at the device level. By default, the encryption service is disabled and passwords are stored in clear text. The following rules apply to password encryption:

- When you enable password encryption, all existing clear-text passwords will be encrypted, and any password that are added subsequently in clear-text are stored in encrypted format

  In the following example, the testuser account password is created in clear text after password encryption has been enabled. The global encryption policy overrides command-level encryption settings The password is stored as encrypted.

```
switch(config)# service password-encryption

switch(config)# do show running-config service password-encryption
service password-encryption

switch(config)# username testuser role testrole desc "Test User" encryption-level 0 password hellothere

switch(config)# do show running-config username
username admin password "BwrsDbB+tABWGWpINOVKoQ==\n" encryptionlevel 7 role admin desc Administrator
username testuser password "cONW1RQ0nTV9Az42/9uCQg==\n" encryption-level 7 role testrole desc "Test User"
username user password "BwrsDbB+tABWGWpINOVKoQ==\n" encryptionlevel 7 role user desc User
```

- When you disable the password encryption service, any new passwords added in clear text will be stored as clear text on the switch. Existing encrypted passwords remain encrypted.

  In the following example, the testuser account password is stored in clear text after password encryption has been disabled. The default accounts, "user" and admin" remain encrypted.

```
switch(config)# no service password-encryption

switch(config)# do show running-config service password-encryption no service password-encryption

switch(config)# username testuser role testrole desc "Test User" encryption-level 0 password hellothere enable
true

switch(config)# do show running-config username
username admin password "BwrsDbB+tABWGWpINOVKoQ==\n" encryptionlevel 7 role admin desc Administrator
username testuser password hellothere encryption-level 0 role testrole desc "Test User"
username user password "BwrsDbB+tABWGWpINOVKoQ==\n" encryptionlevel 7 role user desc User
```

Account lockout policy

The account lockout policy disables a user account when the user exceeds a configurable number of failed login attempts. A user whose account has been locked cannot log in. SSH login attempts that use locked user credentials are denied without the user being notified of the reason for denial.

The account remains locked until explicit administrative action is taken to unlock the account. A user account cannot be locked manually. An account that is not locked cannot be unlocked.

Failed login attempts are tracked on the local switch only. in VCS mode, the user account is locked only on the switch where the lockout occurred; the same user can still try to log in on another switch in the fabric.

The account lockout policy is enforced across all user accounts except for the root account and accounts with the admin role.

Denial of service implications

The account lockout mechanism may be used to create a denial of service (DOS) condition when a user repeatedly attempts to log in to an account by using an incorrect password. Selected privileged accounts, such as root and admin, are exempted from the account lockout policy to prevent these accounts from being locked out by a DOS attack. However these privileged accounts may then become the target of password-guessing attacks.

> ATTENTION
> Brocade advises that you periodically examine the Security Audit logs to determine if such attacks are attempted. Refer to Logging and analyzing security events on page 28.

Password interaction with remote AAA servers

The password policies apply to local switch authentication only. External AAA servers such as RADIUS, TACACS+, or LDAP provide server-specific password-enforcement mechanisms. The password management commands operate on the device-local password database only, even when the device is configured to use an external AAA service for authentication. When so configured, authentication through remote servers is applied to the login only.

When remote AAA server authentication is enabled, an administrator can still perform user and password management functions on the local password database.

For more information on remote AAA server authentication, refer to Managing User Accounts on page 13.

Configuring password policies

Use the **password-attributes** command with specified parameters to define or modify existing password policies.

Configuring the account lockout threshold

You can configure the lockout threshold with the **password-attributes max-retry** *maxretry* command. The value of the *maxretry* specifies the number of times a user can attempt to log in with an incorrect password before the account is locked. The number of failed login attempts is counted from the last successful login. The *maxretry* can be set to a value from 0 through 16. A value of 0 disables the lockout mechanism (default).

The following example sets the lockout threshold to 5.

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.
2. Enter the **password-attributes** command with the specified parameter.

```
switch# configure terminal
Entering configuration mode terminal
switch(config)# password-attributes max-retry 4
```

When a user account is locked, it can be unlocked using the procedure described in Unlocking a user account on page 15.

Creating a password policy

The following example defines a password policy that places restrictions on minimum length and enforces character restrictions and account lockout.

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.

2. Enter the **password-attributes** command with the specified parameters.

```
switch# configure terminal
Entering configuration mode terminal
switch(config)# password-attributes min-length 8 max-retry 4 character-restriction lower 2 upper 1
numeric 1 special-char 1  max-lockout-duration 5000
```

Restoring the default password policy

Entering the **no** form of the **password-attributes** command resets all password attributes to their default values. If you specify a specific attribute, only that attribute is reset to the default. If you enter **no password-attributes** without operands, all password attributes are reset to their default values.

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.
2. Enter the **password-attributes** command with the specified parameters.

```
switch# configure terminal
Entering configuration mode terminal
switch(config)# no password-attributes min-length
switch(config)# password-attributes max-retry 4
switch(config)# no password-attributes numeric
```

Displaying password attributes

To display configured password attributes, switch to privileged EXEC mode and enter **show running-config password-attributes**. Refer to the password-attributes command in the *Network OS Command Reference* for details on modifying password attributes.

```
switch# show running-config password-attributes
password-attributes max-retry 4
password-attributes character-restriction upper 1
password-attributes character-restriction lower 2
password-attributes character-restriction numeric 1
password-attributes character-restriction special-char 1
password-attributes max-lockout-duration 5000
```

Understanding and managing role-based access control

The software uses role-based access control (RBAC) as the authorization mechanism. You can create roles dynamically and associate them with rules to define the permissions applicable to a particular role. Every user account must be associated with a role, and only a single role can be associated with any given account.

RBAC specifies access rights to resources. When a user executes a command, privileges are evaluated to determine access to the command based on the role of the user.

In logical chassis cluster mode, the configuration is applied to all nodes in the cluster.

Default roles

All Brocade devices support two default roles, "user" and "admin." You cannot modify the attributes of default roles; however, you can assign the default roles to non-default user accounts. The default roles have the following access privileges:

- The user role has limited privileges that are restricted to executing **show** commands in privileged EXEC mode, as well as the following operational commands: **ping**, **ssh**, **telnet**, **timestamp**, **rasman**, and **traceroute**. User accounts associated with the user role cannot access configuration commands that are available only in global configuration mode.
- The admin role has the highest privileges. All commands available in privileged EXEC mode and in global configuration mode are accessible to the user associated with the admin role.

With a new switch, only the admin user account has access to perform user and role management operations. The admin user can create any roles and configure those roles for access to user and role management operations.

User-defined roles

In addition to the default roles, Network OS supports the creation of user-defined roles. A user-defined role starts from a basic set of privileges which are then refined by adding special rules. When you have created a role, you can assign a name to the role and then associate the role to one or more user accounts.

The following tools are available for managing user-defined roles:

- The **role** command defines new roles and deletes user-defined roles.
- The **rule** command allows you to specify access rules for specific operations and assign these rules to a given role.
- The **username** command associates a given user-defined role with a specific user account.

A user-defined role has a mandatory name and an optional description, as shown in the following table.

TABLE 3 Role attributes

Parameter	Description
name	The role name must be unique, begin with a letter, and can contain alphanumeric characters and underscores. The length of the role name should be between 4 and 32 characters. The name cannot be same as that of an existing user, an existing default role, or an existing user-defined role.
desc	An optional description of the role. The description can be up to 64 characters and can include any printable ASCII character, except for the following characters: single quotation marks (‘), double quotation marks(“), exclamation point (!), colon (:), and semi-colon (;). If the description contains spaces, you must enclose the text in double quotation marks. if the description contains spaces

The operation of creating a role must satisfy the following criteria to succeed:

- The maximum number of roles supported on a chassis is 64.
- The **role** command must be run from an account authorized for the operation.
- The **role** command is available in global configuration mode.
- If the role specified already exists, the **role** command modifies the existing role.

Displaying a role

In privileged EXEC mode, enter the **show running-config role** command.

```
switch# show running-config role
role name VLANAdmin desc "Manages security CLIs"
role name NetworkAdmin desc "Manages Network CLIs"
role name ClusterAdmin desc "Manages Cluster CLIs"
```

Creating or modifying a role

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.

   ```
   switch# configure terminal
   Entering configuration mode terminal
   ```

2. Enter the **role** command with the specified parameters.

   ```
   switch(config)# role name VLANAdmin desc "Manages security CLIs"
   ```

Deleting a role

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.

   ```
   switch# configure terminal
   Entering configuration mode terminal
   ```

2. Enter the **no role** command with the specified parameters.

   ```
   switch(config)# no role name VLANAdmin
   ```

Commonly used roles

The following examples illustrate the creation and configuration of two frequently-used administrative roles and accounts: Brocade VCS Fabric security administrator, and FCoE Fabric administrator.

Creating a VCS Fabric security administrator role and account

The following steps create and configure a typical Brocade VCS Fabric security administrator role.

1. Create a role for a Brocade VCS Fabric security administrator.

   ```
   switch(config)# role name NetworkSecurityAdmin desc "Manages security CLIs"
   ```

2. Create a user account associated with the newly created role.

   ```
   switch(config)# username SecAdminUser role NetworkSecurityAdmin password testpassword
   ```

3. Create the rules to specify the RBAC permissions for the NetworkSecurityAdmin role.

```
switch(config)# rule 30 action accept operation read-write role NetworkSecurityAdmin command role
switch(config-rule-30)# exit
switch(config)# rule 31 action accept operation read-write role NetworkSecurityAdmin command rule
switch(config-rule-31)# exit
switch(config)# rule 32 action accept operation read-write role NetworkSecurityAdmin command username
switch(config-rule-32)# exit
switch(config)# rule 33 action accept operation read-write role    NetworkSecurityAdmin command aaa
switch(config-rule-33)# exit
switch(config)# rule 34 action accept operation read-write role    NetworkSecurityAdmin command
radius-server
switch(config-rule-34)# exit
switch(config)# rule 35 action accept operation read-write role    NetworkSecurityAdmin command
config
switch(config-rule-35)# exit
```

The SecAdminUser account has been granted operational access to the configuration-level commands **role**, **rule**, **username**, **aaa**, and **radius-server**. Any account associated with the NetworkSecurityAdmin role can now create and modify user accounts, manage roles, and define rules. In addition, the role permits configuring a RADIUS server and set the login sequence.

Creating an FCoE administrator role and account

The following steps create and configure a typical FCoE administrator role.

1. Create an FCoE administrator role.

```
switch(config)# role name FCOEAdmin desc "Manages FCOE"
```

2. Create an FCoE admin user account.

```
switch(config)# username FCOEAdmUser role FCOEAdmin password testpassword
```

3. Create the rules defining the access permissions for the FCoE administrator role.

```
switch(config)# rule 40 action accept operation read-write role FCOEAdmin    command interface fcoe
```

The FCOEAdmUser account that is associated with the FCoEAdmin role can now perform the FCoE operations.

Understanding and managing command access rules

Command authorization is defined in terms of an ordered set of rules that are associated with a role. Rules define and restrict a role to access modes (*read-only* or *read-write* access), and beyond that can define permit or reject on specified command groups or individual commands. You can associate multiple rules with a given user-defined role, but you can associate only one role with any given user account.

To specify a rule, you must specify at least three mandatory attributes: a rule index number, the role to which the rule should apply, and the command that is defined by the rule. The following table describes the rule attribute details.

TABLE 4 Command access rule attributes

Parameter	Description
index	A numeric identifier of the rule in the range between 1 and 512.
role	The name of the role for which the rule is defined.
command	The command for which access is defined.
operation	Optional. Defines the general access mode granted by the rule. Access can be **read-only** or **read-write** (default).

TABLE 4 Command access rule attributes (continued)

Parameter	Description
action	Optional. A modifier restricting the general access mode. The specified access is either accepted (**accept**) or rejected (**reject**). The default value is **accept** .

Specifying rule commands with multiple options

Commands consisting of multiple words indicating command hierarchy are separated by a space, as shown in the following examples.

```
switch(config)# rule 70 action accept operation read-write role NetworkAdmin command copy running-config
switch(config)# rule 71 action accept operation read-write role NetworkAdmin command interface management
switch(config)# rule 72 action accept operation read-write role NetworkAdmin command clear logging
```

NOTE

Rules cannot be added for commands that are not at the top level of the command hierarchy. For a list of eligible commands, type the help function (**?**) at the command prompt.

Verifying rules for configuration commands

The following rules govern configuration commands:

- If a role has a rule with a **read-write** operation and the **accept** action for a configuration command, the user associated with this role can execute the command and read the configuration data.
- If a role has a rule with a **read-only** operation and the **accept** action for a configuration command, the user associated with this role can only read the configuration data of the command.
- If a role has a rule with a **read-only** or **read-write** operation and the **reject** action for a configuration command, the user associated with this role cannot execute the command and can read the configuration data of the command.

Configuring rules for operational commands

Rules can be created for the specified operational commands. By default, every role can display all the operational commands but cannot execute them. The **show** commands can be accessed by all the roles.

The following rules govern operational commands:

- If a role has a rule with a **read-write** operation and the **accept** action for an operational command, the user associated with this role can execute the command.
- If a role has a rule with a **read-only** operation and the **accept** action for an operational command, the user associated with this role can access but cannot execute the command.
- If a role has a rule with a **read-only** or **read-write** operation and the **reject** action for an operational command, the user associated with this role can neither access nor execute the command.

Configuring rules for interface key-based commands

By default, every role has the permission to read the configuration data related to all the instances of the interfaces using the **show running-config interface** *interface_name rbridge-id/slot/port* command.

Rules can be created for a specific instance of the interface-related configuration commands.

The following rules govern interface key-based commands:

- If a role has a rule with a **read-write** operation and the **accept** action for only a particular instance of the interface, the user associated with this role can only modify the attributes of that instance.
- If a role has a rule with a **read-only** operation and the **accept** action for only a particular instance of the interface, the user associated with this role can only read (using the **show running-config** command) the data related to that instance of the interface.
- If a role has a rule with a **read-write** operation and the **reject** action for only a particular instance of the interface, the user associated with this role cannot execute and read the configuration data for that interface instance.

  In the following example, the rules are applicable only to a particular instance of the specified interface.

  ```
  switch(config)# rule 60 action accept operation read-write role NetworkAdmin command interface
  tengigabitethernet 1/0/4
  switch(config)# rule 65 action accept operation read-write role NetworkAdmin    command interface
  fcoe 1/0/4
  switch(config)# rule 68 role NetworkAdmin action reject command interface    fortygigabitethernet
  1/2/4
  ```

- If a role has a rule with a **read-only** or **read-write** operation and the **reject** action for an interface or an instance of the interface, the user associated with this role cannot perform **clear** and **show** operations related to those interfaces or interface instances. To perform **clear** and **show** operations, the user's role must have at least **read-only** and the **accept** permission. By default, every role has the **read-only** and **accept** permission for all interface instances.

  In the following example, the user associated with the NetworkAdmin role cannot perform **clear** and **show** operations related to all **tengigabitethernet** instances.

  ```
  switch(config)# rule 30 action accept operation read-write role NetworkAdmin    command interface
  tengigabitethernet
  ```

- If a role has a rule with **read-only** or **read-write** operation, and the **reject** action for an interface **tengigabitethernet** and **fcoe** instances, the user associated with this role cannot perform **clear** and **show** operations related to those instances. To perform **clear** and **show** operations related to **interface tengigabitethernet** and **fcoe** instances, the user's role should have at least **read-only** and **accept** permission. By default, every role has the **read-only** or **accept** permission for all interface instances.

  In the following example, the user associated with the NetworkAdmin role cannot perform some of the **clear** and **show** operations related to all **tengigabitethernet** instances.

  ```
  switch(config)# rule 30 role NetworkAdmin action reject command interface    tengigabitethernet
  ```

- A rule created with the **no-operation** command does not enforce any authorization rules. Instead, the **no-operation** instance can be considered as a placeholder for a valid command that will be added later.

  ```
  switch(config)# rule 75 action reject operation read-write role NetworkAdmin    command no-operation
  switch(config)# rule 75 command firmware
  ```

- The **dot1x** option under the **interface** instance submode can only be configured if the role has the **read-write** and **accept** permissions for both the **dot1x** command and **interface te** instances.

  In the following example, the user associated with the CfgAdmin role can access and execute the **dot1x** command in the specified **tengigabitethernet** instance.

  ```
  switch(config)# rule 16 action accept operation read-write role cfgadmin    command interface
  tengigabitethernet
  switch(config)# rule 17 action accept operation read-write role cfgadmin    command dot1x
  ```

- To execute the **no vlan** and **no spanning-tree** commands under the submode of **interface tengigabitethernet** instances, a user must have **read-write** and **accept** permissions for both the **vlan** and the **protocol spanning-tree** commands. If a user has **read-**

write and **accept** permissions for the **vlan** and **spanning-tree** commands and **read-write** and **accept** permissions for at least one interface instance, the user can perform the **no vlan** and **no spanning-tree** operations on the other **interface** instances for which the user has only default permissions (**read-only** and **accept**).

Configuring a placeholder rule

A rule created with the **no-operation** command does not enforce any authorization rules. Instead, you can use the **no-operation** instance as a placeholder for a valid command that is added later, as shown in the following example.

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.

```
switch# configure terminal
Entering configuration mode terminal
```

2. Enter the **rule** command with the specified parameters and the **no-operation** keyword as a placeholder.

```
switch(config)# rule 75 action reject operation read-write role NetworkAdmin command no-operation
```

3. Enter the **rule** command with the specified command to replace the placeholder.

```
switch(config)# rule 75 command firmware
```

Configuring rule processing

When a user executes a command, rules are searched in ascending order by index for a match and the action of the first matching rule is applied. If none of the rules match, command execution is blocked. If there are conflicting permissions for a role in different indices, the rule with lowest index number is applied.

As an exception, when a match is found for a rule with the **read-only** operation and the **accept** action, the system seeks to determine whether there are any rules with the **read-write** operation and the **accept** action. If such rules are found, the rule with the **read-write** permission is applied.

In the following example, two rules with action accept are present and rule 11 is applied.

```
switch(config)# rule 9 operation read-only action accept role NetworkAdmin command aaa
switch(config)# rule 11 operation read-write action accept role NetworkAdmin command aaa
```

Adding a rule

You add a rule to a role by entering the **rule** command with appropriate options. Any updates to the authorization rules will not apply to the active sessions of the users. The changes are applied only when users log out from the current session and log in to a new session.

The following example creates the rules that authorize the security administrator role to create and manage user accounts:

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.

```
switch# configure terminal
Entering configuration mode terminal
```

2. Create a rule specifying read-write access to the global configuration mode.

```
switch(config)# rule 150 action accept operation read-write role SecAdminUser    command config
```

3. Create a second rule specifying read-write access to the **username** command. Enter the **rule** command with the specified parameters.

```
switch(config)# rule 155 action accept operation read-write role SecAdminUser command username
```

4. After creating the rules, the user of the SecAdminUser account can log in to the switch and create or modify the user accounts by using the **username** command.

```
switch login: SecAdminUser
Password:******
Welcome to the ConfD CLI
SecAdminUser connected from 127.0.0.1 using console on switch

switch# configure terminal
Entering configuration mode terminal
Current configuration users:
admin console (cli from 127.0.0.1) on since 2010-08-16 18:35:05 terminal mode

switch(config)# username testuser role user password (<string>): ********
```

Changing a rule

The following example changes the previously created rule (index number 155) so that the **username** command is replaced by the **role** command.

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.

```
switch# configure terminal
Entering configuration mode terminal
```

2. Enter the **rule** command, specifying an existing rule (index 155) and changing the **command** attribute to the **role** command.

```
switch(config)# rule 155 command role
```

After changing rule 155, SecAdminUser can log in to the switch and execute the **role** command and not the **username** command.

```
switch# login SecAdminUser
switch# Password: ***********
Welcome to the ConfD CLI
SecAdminUser connected from 127.0.0.1 using console on sw0
switch# configure terminal
Entering configuration mode terminal
Current configuration users:
admin console (cli from 127.0.0.1) on since 2010-08-16 18:35:05 terminal mode
switch(config)# role name NetworkAdmin
```

Deleting a rule

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.

```
switch# configure terminal
Entering configuration mode terminal
```

2. Enter the **no rule** command followed by the index number of the rule you wish to delete.

```
switch(config)# no rule 155
```

After rule 155 is deleted, the SecAdminUser can no longer access the **role** command.

Displaying a rule

Enter the **show running-config rule** command in privileged EXEC mode to display all configured rules. You can modify the output by using the command and specifying additional parameters.

```
switch# show running-config rule
rule 30 action accept operation read-write role NetworkSecurityAdmin rule 30 command role
rule 31 action accept operation read-write role NetworkSecurityAdmin rule 31 command rule
rule 32 action accept operation read-write role NetworkSecurityAdmin rule 32 command username
rule 33 action accept operation read-write role NetworkSecurityAdmin rule 33 command aaa
rule 34 action accept operation read-write role NetworkSecurityAdmin rule 34 command radius-server
rule 35 action accept operation read-write role NetworkSecurityAdmin rule 35 command configure
rule 40 action accept operation read-write role FCOEAdmin rule 40 command "interface fcoe"
```

Logging and analyzing security events

Security event logging utilizes the RASLog audit infrastructure to record security-related audit events. Any user-initiated security event generates an auditable event. Audited events are generated for all Management interfaces. In Brocade VCS Fabric mode, for cluster-wide events, the audit is generated on all switches of the cluster.

Refer to the *Network OS Message Reference* for information on how to configure, monitor, and analyze security audit logging.

Configuring External Server Authentication

Understanding and configuring remote server authentication

Remote server authentication overview

The software supports various protocols to provide external Authentication, Authorization, and Accounting (AAA) services for Brocade devices. Supported protocols include the following:

- RADIUS — Remote authentication dial-in user service
- LDAP/AD — Lightweight Directory Access Protocol using Microsoft Active Directory (AD) in Windows
- TACACS+ — Terminal access controller access-control system plus

When configured to use a remote AAA service, the switch acts as a network access server client. The switch sends all authentication, authorization, and accounting (AAA) service requests to the remote RADIUS, LDAP, or TACACS+ server. The remote AAA server receives the request, validates the request, and sends a response back to the switch.

The supported management access channels that integrate with RADIUS, TACACS+, or LDAP include serial port, Telnet, or SSH.

When configured to use a remote RADIUS, TACACS+, or LDAP server for authentication, a switch becomes a RADIUS, TACACS+, or LDAP client. In either of these configurations, authentication records are stored in the remote host server database. Login and logout account name, assigned permissions, and time-accounting records are also stored on the AAA server for each user.

Brocade recommends that you configure at least two remote AAA servers to provide redundancy in the event of failure. For each of the supported AAA protocols, you can configure up to five external servers on the switch. Each switch maintains its own server configuration.

Login authentication mode

The authentication mode is defined as the order in which AAA services are used on the switch for user authentication during the login process. The software supports two sources of authentication: primary and secondary. The secondary source of authentication is used in the event of primary source failover and is optional for configuration. You can configure four possible sources for authentication:

- Local — Use the default switch-local database (default)
- RADIUS — Use an external RADIUS server
- LDAP — Use an external LDAP server
- TACACS+ — Use an external TACACS+ server

By default, external AAA services are disabled, and AAA services default to the switch-local user database. Any environment requiring more than 64 users should adopt AAA servers for user management.

When the authentication, authorization, and accounting (AAA) mode is changed, an appropriate message is broadcast to all logged-in users, and the active login sessions end. If the primary source is set to an external AAA service (RADIUS, LDAP, or TACACS+) and the secondary source is not configured, the following events occur:

- For Telnet-based and SSH connections-based logins, the login authentication fails if none of the configured (primary source) AAA servers respond or if an AAA server rejects the login.
- For a serial port (console) connection-based login, if a user's login fails for any reason with the primary source, failover occurs and the same user credentials are used for login through the local source. This failover is not explicit.
- If the primary source is set to an external AAA service, and the secondary source is configured to be local (for example, by means of the **aaa authentication login radius local** command), then, if login fails through the primary source either because none of the configured servers is responding or the login is rejected by a server, failover occurs and authentication occurs again through the secondary source (local) for releases earlier than Network OS 4.0.

In Network OS 4.0 and later, when **local** is specified as the secondary authentication service, failover to local does not occur if login is rejected by a server. In addition, when the authentication service is changed, the user sessions are not logged out. If a user wants to log out all connected user sessions, the **clear sessions** command should be used.

- In Network OS 4.0 and later, when **local** is specified as the secondary authentication service, local authentication is tried only when the primary AAA authentication service (TACACS+, RADIUS, or LDAP) is either unreachable or not available. Local authentication will not be attempted if authentication with the primary service fails.
- In Network OS 4.0 and later, you can specify to use the local switch database if prior authentication methods on a RADIUS or TACACS+ server are not active or if authentication fails. To specify this option, use the **local-auth-failback** command. In the following example, the local switch database will be used if the RADIUS server is unavailable.

```
switch(config)# aaa authentication login radius local-auth-fallback
```

Conditions for conformance

- If the first source is specified as **default**, do not specify a second source. A second source signals a request to set the login authentication mode to its default value, which is **local**. If the first source is **local**, the second source cannot be set to any value, because the failover will never occur.
- The source of authentication (except **local**) and the corresponding server type configuration are dependent on each other. Therefore, at least one server should be configured before that server type can be specified as a source.
- If the source is configured to be a server type, you cannot delete a server of that type if it is the only server in the list. For example, if there are no entries in the TACACS+ server list, the authentication mode cannot be set to **tacacs+** or **tacacs+ local**. Similarly, when the authentication mode is **radius** or **radius local**, a RADIUS server cannot be deleted if it is the only one in the list.

Configuring remote server authentication

This section introduces the basics of configuring remote server authentication using RADIUS and TACACS+.

- Understanding and configuring RADIUS on page 43
- Understanding and configuring TACACS+ on page 53
- Understanding and configuring LDAP on page 33

Setting and verifying the login authentication mode

The following procedure configures TACACS+ as the primary source of authentication and the switch-local user database as the secondary source. For complete information on login authentication mode, refer to the **aaa authentication login** command in the *Network OS Command Reference*.

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.

```
switch# configure terminal
Entering configuration mode terminal
```

2. Enter the **aaa authentication login** command with the specified parameters.

```
switch(config)# aaa authentication login tacacs+ local

Broadcast message from root (pts/0) Tue Apr 5 16:34:12 2011...
AAA Server Configuration Change: all accounts will be logged out
```

3. Enter the **do show running-config aaa** command to display the configuration.

```
switch(config)# do show running-config aaa
aaa authentication login tacacs+ local
```

4. Log in to the switch using an account with TACACS+-only credentials to verify that TACACS+ is being used to authenticate the user.

Resetting the login authentication mode

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.

```
switch# configure terminal
Entering configuration mode terminal
```

2. Enter the **no aaa authentication login** command to remove the configured authentication sequence and to restore the default value (Local only).

```
switch(config)# no aaa authentication login
```

3. Verify the configuration with the **do show running-config aaa** command.

```
switch(config)# do show running-config aaa
aaa authentication login local
```

4. Log in to the switch using an account with TACACS+-only credentials. The login should fail with an "access denied" error.
5. Log in to the switch using an account with local-only credentials. The login should succeed.

Changing the login authentication mode

You can set the authentication mode with the **aaa authentication login** command, but you cannot change or delete an existing authentication mode with the same command. You can only reset the configuration to the default value using the **no aaa authentication login** command and then reconfigure the authentication sequence to the correct value.

NOTE

In a configuration with primary and secondary sources of authentication, the primary mode cannot be modified alone. First remove the existing configuration and then configure it to the required configuration.

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.

```
switch# configure terminal
Entering configuration mode terminal
```

2. Enter the **no aaa authentication login** command to reset the configuration to the default value.

```
switch(config)# no aaa authentication login tacacs+ local
```

3. Enter the **aaa authentication login** command and specify the desired authentication mode.

```
switch(config)# aaa authentication login radius local
Broadcast message from root (pts/0) Tue Apr 5 16:34:12 2011...
AAA Server Configuration Change: all accounts will be logged out
```

4. Verify the configuration with the **do show running-config aaa** command.

```
switch(config)# do show running-config aaa
aaa authentication login radius local
```

5. Log in to the switch using an account with TACACS+ credentials. The login should fail with an "access denied" error.
6. Log in to the switch using an account with RADIUS credentials. The login should succeed.

LDAP Server Authentication

Understanding and configuring LDAP

Lightweight Directory Access Protocol (LDAP) is an open-source protocol for accessing distributed directory services that act in accordance with X.500 data and service models. LDAP assumes that one or more servers jointly provide access to a Directory Information Tree (DIT) where data is stored and organized as entries in a hierarchical fashion. Each entry has a name called the distinguished name that uniquely identifies it.

LDAP can also be used for centralized authentication through directory service.

Active Directory (AD) is a directory service which supports a number of standardized protocols such as LDAP, Kerberos authentication, and DNS, to provide various network services. AD uses a structured data store as the basis for a logical, hierarchical organization of directory information. AD includes user profiles and groups as the part of directory information, so it can be used as a centralized database for authenticating the third-party resources.

If you are in logical chassis cluster mode, the configuration is applied to all nodes in the cluster.

User authentication

A Brocade device can be configured as an LDAP client for authentication with an Active Directory (AD) server, supporting authentication with a clear text password over the Transport Layer Security (TLS) channel. Optionally, it supports server authentication during the TLS handshake. Only the user principal name from the AD server is supported for LDAP authentication on the Brocade device. The Common Name-based authentication is not supported. When you log in from the device, the complete user principal name, including domain, should be entered (for example, "testuser@sec.example.com").

LDAP supports alternative user principal names, such as:

- username
- username@AD.com
- username@ADsuffix.com
- username@newUPN.com

Network OS supports LDAP authentication with the following AD servers:

- Windows 2000
- Windows 2003
- Windows 2008 AD

A Brocade device configured to perform LDAP-based authentication supports its access through a serial port, Telnet, and SSH. These access channels require that you know the device IP address or name to connect to the devices.

A maximum of five AD servers can be configured on a Brocade device.

If you are in logical chassis cluster mode, all LDAP server and map role configurations (except "show certutil" and "certutil") are applied to all devices in the cluster.

Server authentication

As a part of user authentication using LDAP, the Brocade device can be configured to support server certificate authentication. To enable server authentication (server certificate verification), follow these guidelines:

- While configuring the LDAP server, the Fully Qualified Domain Name (FQDN) of the AD server should be added as the host parameter, instead of the IP address. A FQDN is needed to validate the server identity as mentioned in the common name of the server certificate.
- The DNS server must be configured on the device prior to adding AD server with a domain name or a hostname. Without a DNS server, the name resolution of the server fails, and then the add operation fails. Use the **ip dns** command to configure DNS.
- The CA certificate of the AD server's certificate should be installed on the device. Currently, only PEM-formatted CA certificates can be imported into the device.

If more than one server is configured and an LDAP CA certificate is imported for one server on the device, the device performs the server certificate verification on all servers. Thus, either CA certificates for all servers should be imported, or CA certificates should not be imported for any of the servers. After the CA certificate is imported, it is retained even if the device is set back to its default configuration. If the CA certificate is not required, you should explicitly delete it.

Server authorization

The Active Directory (AD) server is used only for authentication. Command authorization of the AD users is not supported in the AD server. Instead, the access control of AD users is enforced locally by role-based access control (RBAC) on the device.

A user on an AD server should be assigned a nonprimary group, and that group name should be either matched or mapped to one of the existing roles on the device; otherwise, authentication will fail. After successful authentication, the device receives the nonprimary group of the user from the AD server and finds the corresponding user role for the group based on the matched or mapped roles.

If the device fails to get the group from the AD server, or the LDAP user is not a member of any matching AD group, the user authentication fails. Groups that match with the existing device roles have higher priority than the groups that are mapped with the device roles. Thereafter, the role obtained from AD server (or default role) is used for RBAC.

If multiple nonprimary groups are associated to the AD user, only one of the groups should be mapped or matched to the device role. If multiple AD groups of AD users are mapped or matched to the device roles, authentication of the user is successful, but there is no guarantee as to which role the AD user gets among those multiple roles. After successful authentication, the device gets the nonprimary group of the user from the AD server and finds the corresponding user role for group based on the matched or mapped roles. Thereafter, the role obtained from the AD server (or default role) will be used for RBAC.

A maximum 16 AD groups can be mapped to the device roles.

FIPS compliance

To support FIPS compliance, the CA certificate of the AD server's certificate should be installed on the switch, and the FIPS-compliant TLS ciphers for LDAP should be used.

Configuring LDAP

Configuring support for LDAP requires configuring both the client and the server. This section presents the following major tasks, sorted by client-side and server-side activities:

Client-side tasks:

- Configuring an Active Directory server on the client side on page 35
- Configuring Active Directory groups on the client side on page 39
- Clearing sessions on the client side on page 40

Server-side tasks:

- Configuring an Active Directory server on the client side on page 35

Importing an LDAP CA certificate

The following example imports the LDAP CA certificate from a remote server to a Brocade switch using secure copy (SCP).

1. In privileged EXEC mode, enter **configure terminal** to change to global configuration mode.

```
switch# configure terminal
Entering configuration mode terminal
```

2. Enter **certutil import ldapca** with the specified parameters.

```
switch# certutil import ldapca directory /usr/ldapcacert file cacert.pem protocol SCP host
10.23.24.56 user admin password *****
```

3. Verify the import by entering **show cert-util ldapcacert**.

```
switch# show cert-util ldapcacert
List of ldap ca certificate files:
swLdapca.pem
```

Deleting LDAP CA certificates

The **no certutil ldapca** command deletes the LDAP CA certificates of all Active Directory servers. You must confirm that you want to delete the certificates.

```
switch# no certutil ldapca
Do you want to delete LDAP CA certificate? [y/n]:y
```

Configuring an Active Directory server on the client side

Each Brocade switch client must be individually configured to use Active Directory servers. You use the **ldap-server** command to specify the host server, authentication protocols, and other parameters. You can configure a maximum of five Active Directory servers on a Brocade switch for AAA service.

The parameters in the following table are associated with an Active Directory server that is configured on the switch.

TABLE 5 Active Directory parameters

Parameter	Description
host	IP address (v4) or Fully Qualified Domain name of the AD server. IPv6 is supported for Windows 2008 AD server only. The maximum supported length for the host name is 40 characters.
port	TCP port used to connect the AD server for authentication. The valid port range is 1024 through 65535. The default port is 389.
timeout	Time to wait for a server to respond. The range is 1 through 60 seconds. The default value is 5 seconds.
retries	Number of unsuccessful attempts to be made to connect to an AD server before quitting. The valid range is 1 through 100. The default value is 5.
domain	Base domain name

A maximum of five LDAP/AD servers can be configured on a Brocade switch for authentication service.

Adding an LDAP server to the client server list

The following procedure configures an LDAP server on an ADAP client (Brocade switch).

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.

```
switch# configure terminal
Entering configuration mode terminal
```

2. Use the **ldap-server-host** command to set the parameters for the LDAP server.

   This command places you into the LDAP server configuration submode where you can modify the server default settings.

```
switch(config)# ldap-server host 10.24.65.6 basedn sec.brocade.com port 3890
switch(config-ldap-server-10.24.65.6)#
```

3. Modify any settings, such as the domain name or retry limit, in this configuration mode (refer to the preceding table).

```
switch(config-ldap-server 10.24.65.6)# basedn security.brocade.com
switch(config-ldap-server 10.24.65.6)# timeout 8
switch(config-host-10.24.65.6)# retries 3
```

4. Confirm the LDAP settings with the **do show running-config ldap-server** command.

   Attributes holding default values are not displayed.

```
switch(config-ldap-server-10.24.65.6)# do show running-config ldap-server host 10.24.65.6

ldap-server host 10.24.65.6
port      3890
basedn    security.brocade.com
retries   3
timeout   8
!
```

5. Use the **exit** command to return to the global configuration mode.

```
switch(config-ldap-server-10.24.65.6)# exit
```

6. Use the **no ldap-server** command to set an attribute back to the default value.

```
switch(config)# no ldap-server host 10.24.65.6 retries
```

Changing LDAP server parameters

Changing an LDAP server follows the same procedure as that noted for adding an LDAP server to the client server list. Enter the host IP address or host name, then enter the new values as required. Refer to Adding an LDAP server to the client server list on page 36.

```
switch# configure terminal
Entering configuration mode terminal
switch(config)# ldap-server host 10.24.65.6
switch(config-host-10.24.65.6)# domain security.brocade.com
```

Removing an LDAP server

The following example deletes an LDAP server entry from the switch LDAP server list.

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode

```
switch# configure terminal
Entering configuration mode terminal
```

2. Use the **no ldap-server** command to delete the LDAP server.

```
switch(config)# no ldap-server host 10.24.65.6
```

Importing an LDAP CA certificate

This procedure imports the LDAP CA certificate from the remote host to the switch.

1. Connect to the switch and log in using an account with admin role permissions.
2. In privileged EXEC mode, enter the **certutil import ldapca** command. Include the full path to the certificate on the host, specify SCP as the protocol, and include the IP address of the host.

```
switch# certutil import ldapca directory /usr/ldapcacert/ file cacert.pem protocol SCP host
10.23.24.56 user jane password rbridge-id 3
password: ****
```

Deleting an LDAP CA certificate

This procedure deletes the LDAP CA certificates of all attached Microsoft Active Directory servers from the switch.

1. Connect to the switch and log in using an account with admin role permissions.
2. In privileged EXEC mode, enter the **no certutil ldapca** command.

```
switch# no certutil ldapca
Do you want to delete LDAP CA certificate? [y/n]:y
```

3. Enter **Y** to confirm that you want to delete the LDAP CA certificates.

Verifying LDAP CA certificates

To test whether an LDAP CA certificate has been imported on the switch, in privileged EXEC mode, enter the **no certutil ldapca** command and examine the message returned by the system. The command returns an error if there is no LDAP CA certificate on the switch. If an LDAP CA certificate exists on the switch, you are prompted to delete it. Enter **no certutil ldapcacert** command to retain the certificate.

When no LDAP CA certificate is present

```
switch# no certutil ldapcacert
% Error: LDAP CA certificate does not exist.
```

When an LDAP CA certificate exists on the switch

```
switch# no certutil ldapcacert
List of swLdapca.pem files:
swLdapca.pem
```

Viewing the LDAP CA certificate

The following procedure allows you to view the LDAP CA certificate that has been imported on the switch.

1. Connect to the switch and log in using an account with admin role permissions.
2. In privileged EXEC mode, enter the **certutil import syslogca** command. Include the full path to the certificate on the host, specify SCP as the protocol, and include the IP address of the host.

Logical chassis cluster mode

To view the output in logical chassis cluster mode, enter **show cert-util ldapcacert** followed by the desired RBridge ID. This example displays the certificate for RBridge ID 3.

```
switch# show cert-util ldapcert rbridge-id 3
```

Importing a syslog CA certificate

The following procedure imports the syslog CA certificate from the remote host to the switch.

1. Connect to the switch and log in using an account with admin role permissions.
2. In privileged EXEC mode, enter the **certutil import syslogca** command. Include the full path to the certificate on the host, specify SCP as the protocol, and include the IP address of the host.

Logical chassis cluster mode

```
switch# certutil import syslogca directory /usr/ldapcacert/ file cacert.pem protocol SCP host 10.23.24.56
user jane password rbridge-id 3
password: ****
```

Deleting a syslog CA certificate

The following procedure deletes the syslog CA certificates of all attached Active Directory servers from the switch.

1. Connect to the switch and log in using an account with admin role permissions.
2. In Privileged EXEC mode, enter the **no certutil syslogca** command. You will be prompted to confirm that you want to delete the syslogca certificates.

Logical chassis cluster mode

This example deletes the syslogca certificates for RBridge ID 3 only.

```
switch# no certutil syslogca rbridge-id 3
Do you want to delete syslogca certificate? [y/n]:y
Warning: All the syslog CA certificates are deleted.
```

Verifying syslog CA certificates

To test whether a syslogCA certificate has been imported on the switch, in privileged EXEC mode, enter the **no certutil syslogca** command and examine the message returned by the system. The command returns an error if there is no syslog CA certificate on the switch. If a syslog CA certificate exists on the switch, you are prompted to delete it. Enter the **no certutil syslogcacert** command to retain the certificate.

When no syslog CA certificate is present

```
switch# no certutil syslogcacert
% Error: syslog CA certificate does not exist.
```

When a syslog CA certificate exists on the switch

```
switch# no certutil syslogcacert
Do you want to delete syslog CA certificate? [y/n]:n
```

Viewing the syslog CA certificate

The following procedure allows you to view the syslog CA certificate that has been imported on the switch.

1. Connect to the switch and log in using an account with admin role permissions.
2. In privileged EXEC mode, enter the **show cert-util syslogcacert** command.

Logical chassis cluster mode

This example displays the syslog CA certificates for rbridge-id 3 only.

```
switch# show cert-util syslogcacert rbridge-id 3
```

Configuring Active Directory groups on the client side

An Active Directory (AD) group defines access permissions for the LDAP server similar to Brocade roles. You can map an Active Directory group to a Brocade role with the **ldap-server maprole** command. The command confers all access privileges defined by the Active directory group to the Brocade role to which it is mapped.

A user on an AD server should be assigned a nonprimary group, and that group name should be either matched or mapped to one of the existing roles on the device.

After successful authentication, the user is assigned a role from a nonprimary group (defined on the AD server) based on the matched or mapped device role.

A user logging in to the device that is configured to use LDAP and has a valid LDAP user name and password will be assigned LDAP user privileges if the user is not assigned with any nonprimary group.

Mapping an Active Directory group to a switch role

In the following example, a Brocade user with the admin role inherits all privileges associated with the Active Directory (AD) Administrator group.

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.

   ```
   switch# configure terminal
   Entering configuration mode terminal
   ```

2. Use the **ldap-server maprole** command to set the group information.

   A maximum of 16 AD groups can be mapped to the switch roles.

   ```
   switch(config)# ldap-server maprole group Administrator role admin
   ```

Removing the mapping of an Active Directory to a switch role

The following example removes the mapping between the Brocade admin role and the Active Directory (AD) Administrator group. A Brocade user with the admin role can no longer perform the operations associated with the AD Administrator group.

To unmap an AD group to a device role, perform the following steps from privileged EXEC mode.

1. Use the **configure terminal** command to enter global configuration mode.

```
switch# configure terminal
Entering configuration mode terminal
```

2. Use the **no ldap-server maprole** command to set the group information.

```
switch(config)# no ldap-server maprole group Administrator
```

Configuring the client to use LDAP/AD for login authentication

After you configured the switch LDAP server list, you must set the authentication mode so that ALDAP is used as the primary source of authentication. Refer to Login authentication mode on page 29 for information on how to configure the login authentication mode.

Clearing sessions on the client side

In Network OS 4.0 and later, you can use the **clear sessions** command to log out user sessions that are connected to a device. This command is not distributed across a cluster. If you are in VCS mode, you must use the RBridge ID of the node to log out the users connected to the individual nodes.

```
switch# clear sessions rbridge-id 3
This operation will logout all the user sessions. Do you want to continue (yes/no)?: y
```

Configuring an Active Directory server on the client side

The following high-level overview of server-side configuration for LDAP/AD servers indicates the steps needed to set up a user account. This overview is provided for your convenience only. All instructions involving Microsoft Active Directory can be obtained from www.microsoft.com or from your Microsoft documentation. Confer with your system or network administrator prior to configuration for any special needs your network environment may have.

Creating a user account on an LDAP/AD server

1. Create a user on the Microsoft Active Directory server.
2. Create a group. The group should either match with the user's Brocade switch role or you can map the role to the Brocade switch role with the **ldap-server maprole** command.
3. Associate the user with the group by adding the user to the group.
   The user account configuration is complete.

Verifying the user account on the switch

1. Log in to the switch as a user with admin privileges.
2. Verify that the LDAP/AD server has an entry in the switch LDAP server list.

```
switch# show running-config ldap-server
```

3. In global configuration mode, set the login authentication mode on the switch to use LDAP only and verify the change.

```
switch# configure terminal
Entering configuration mode terminal
switch(config)# no aaa authentication login
switch(config)# aaa authentication login ldap
switch(config)# do
 show running-config aaa
aaa authentication login ldap
```

4. Log in to the switch using an account with valid LDAP/AD only credentials to verify that LDAP/AD is being used to authenticate the user.
5. Log in to the switch using an account with switch-local only credentials. The login should fail with an access denied message.

Configuring LDAP users on an AD server

1. Create a user.
   a) Go to **Programs** > **Administrative Tools** > **Active directory Users and Computers**.
   b) Add a user by completing the **Active directory Users and Computers** dialog box.
   c) Save the account information.
   d) From a command prompt, log in using the new user name and enter a password when prompted.
2. Create a group.
   a) Go to **Programs** > **Administrative Tools** > **Active directory Users and Computers**.
   b) Add a new group.
   c) Save the group information.
3. Assign the group to the user.
   a) Click on the user name.
   b) From the **Properties** dialog box, click the **Member Of** tab and update the field with the group name. This group should either match the switch role or it must be mapped with the switch role on the Brocade switch. In this instance, Domain Users is the primary group and therefore should not be mapped with the switch role.

RADIUS Server Authentication

Understanding and configuring RADIUS

The remote authentication dial-in user service (RADIUS) protocol manages authentication, authorization, and accounting (AAA) services centrally. The supported management access channels that integrate with RADIUS are serial port, Telnet, and SSH.

If you are in logical chassis cluster mode, the configuration is applied to all nodes in the cluster.

Authentication and accounting

When a Brocade device is configured with a set of RADIUS servers to be used for authentication, the device also sends accounting data to the RADIUS server implicitly.

When RADIUS authentication is implemented, the Brocade device consults a RADIUS server to verify user names and passwords. You can optionally configure RADIUS authorization, in which the Brocade device consults a list of commands supplied by the RADIUS server to determine whether a user can issue a command he or she has entered, as well as accounting , which causes the Brocade device to log information on a RADIUS accounting server when specified events occur on the device.

You can use a Remote Authentication Dial In User Service (RADIUS) server to secure the following types of access to the Brocade Layer 2 device or Layer 3 device:

- Telnet access
- SSH access
- Web management access
- Access to the Privileged EXEC level and CONFIG levels of the CLI

The only accounting events supported on Brocade VDX switches configured to use RADIUS are successful login and logout of the RADIUS user.

During the user authentication process, the device sends its IP address. When the device also has a Virtual IP address (in Brocade VCS Fabric mode), it still sends only its unique IP address to the RADIUS server.

> NOTE
> If the RADIUS server is not configured to support accounting, the accounting events sent by the device to the server are dropped.

Authorization

User authorization through the RADIUS protocol is not supported. The access control of RADIUS users is enforced by the Brocade role-based access control (RBAC) protocol at the device level. A RADIUS user should therefore be assigned a role that is present on the device using the Vendor Specific Attribute (VSA) *Brocade-Auth-Role*. After the successful authentication of the RADIUS user, the role of the user configured on the server is obtained. If the role cannot be obtained or if the obtained role is not present on the device, the user will assigned "user" role and a session is granted to the user with "user" authorization.

Account password changes

All existing mechanisms for managing device-local user accounts and passwords remain functional when the device is configured to use RADIUS. Changes made to the device-local database do not propagate to the RADIUS server, nor do the changes affect any account on the RADIUS server; therefore, changes to a RADIUS user password must be done on the RADIUS server.

RADIUS authentication through management interfaces

You can access the device through Telnet or SSH from either the Management interface or the data ports (TE interface or in-band). The device goes through the same RADIUS-based authentication with either access method.

Configuring server-side RADIUS support

With RADIUS servers, you should set up user accounts by their true network-wide identity, rather than by the account names created on a Brocade device. Along with each account name, you must assign appropriate device access roles. A user account can exist on a RADIUS server with the same name as a user on the device at the same time.

When logging in to a device configured with RADIUS, users enter their assigned RADIUS account names and passwords when prompted. Once the RADIUS server authenticates a user, it responds with the assigned device role and information associated with the user account information using a Brocade Vendor-Specific Attribute (VSA). An Authentication-Accept response without the role assignment automatically grants the "user" role.

NOTE
RADIUS requires that you configure both the client and the server.

Configuring a RADIUS server with Linux

FreeRADIUS is an open source RADIUS server that runs on Linux (all versions), FreeBSD, NetBSD, and Solaris. Download the package from www.freeradius.org and follow the installation instructions at the FreeRADIUS website.

You will need the following information to configure Brocade-specific attributes. Refer to the RADIUS product documentation for information on configuring and starting up a RADIUS server.

Adding the Brocade attribute to the RADIUS server configuration

For the configuration on a Linux FreeRADIUS server, define the values outlined in the following table in a vendor dictionary file named dictionary.brocade.

TABLE 6 dictionary.brocade file entries

Include	Key	Value
VENDOR	Brocade	1588

TABLE 6 dictionary.brocade file entries (continued)

Include	Key	Value
ATTRIBUTE	Brocade-Auth-Role	1 string Brocade

1. Create and save the file `$PREFIX/etc/raddb/dictionary.brocade` with the following information:

```
#
# dictionary.brocade
#
VENDOR Brocade 1588
#
# attributes
#
ATTRIBUTE         Brocade-Auth-Role           1      string      Brocade.
```

2. Open the master dictionary file `$PREFIX/etc/raddb/dictionary` in a text editor and add the line:

```
$INCLUDE dictionary.brocade
```

   The file `dictionary.brocade` is located in the RADIUS master configuration directory and loaded for use by the RADIUS server.

Configuring a Brocade user account

When you use network information service (NIS) for authentication, the only way to enable authentication with the password file is to force the Brocade switch to authenticate using password authentication protocol (PAP); this requires the setting the **pap** option with the **radius-server host** command.

1. Open the `$PREFIX/etc/raddb/users` file in a text editor.
2. Add the user name and associated the permissions.

   The user must log in using the permissions specified with Brocade-Auth-Role.

   The following example configures an account called "jsmith" with admin permissions and a password "jspassword".

```
jsmith    Auth-Type := Local,
                  User-Password == "jspassword",
                  Brocade-Auth-Role = "admin"
```

   NOTE
   You must use double quotation marks around the password and role.

Configuring RADIUS server support with a Windows server

Step-by-step instructions for installing and configuring Internet Authentication Service (IAS) with Microsoft Windows server 2008 (or earlier versions, Windows 2003 or 2000) can be obtained from www.microsoft.com or your Microsoft documentation. Confer with your system or network administrator prior to configuration for any special needs your network environment may have.

Use the following information to configure the Internet Authentication Service for a Brocade switch.

NOTE
This is not a complete presentation of steps.

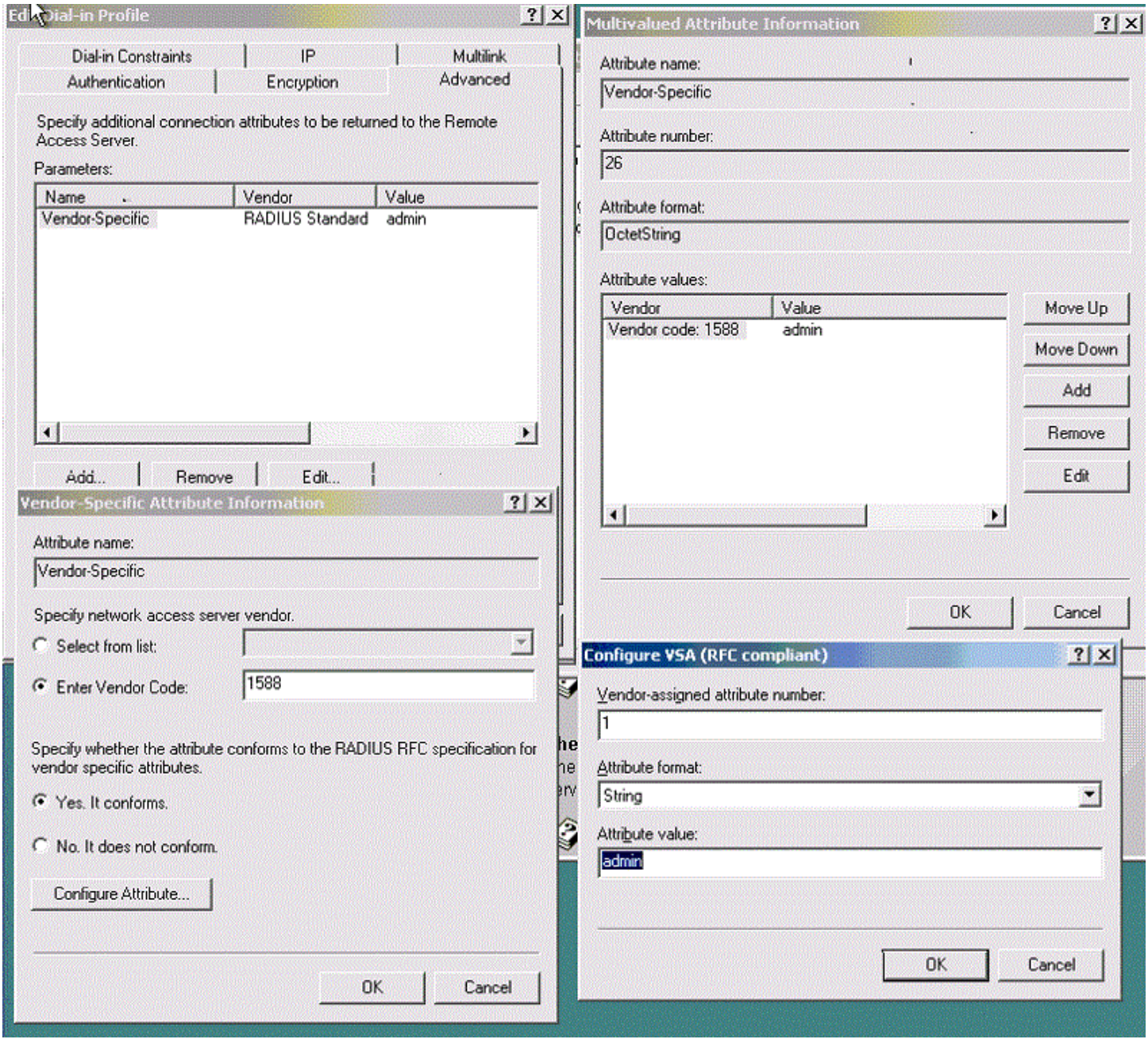
1. In the **New RADIUS Client** window, choose **RADIUS Standard** from the **Client-Vendor** menu.

2. Configure the **Dial-in Profile** dialog box as follows:
   a) Select the **Advanced** tab.
   b) Scroll to the bottom of the RADIUS Standard list, select **Vendor-Specific**, and click **Add**.
   The **Multivalued Attribute Information** dialog box appears.
   c) Click **Add** in the **Multivalued Attribute Information** dialog box.
   The **Vendor-Specific Attribute Information** dialog box appears.
   d) Enter the Brocade vendor code value of **1588**.
   e) Select **Yes. It conforms.** and then click **Configure Attribute**.
   The **Configure VSA (RFC compliant)** dialog box appears.
   f) In the **Configure VSA (RFC compliant)** dialog box, enter the following values and click **OK**:
      - Vendor-assigned attribute number—Enter the value **1**.
      - Attribute format—Enter the value **String**.

   The RADIUS server is now configured.

The following image shows the different screens configured in this task.

FIGURE 1 Windows server VSA configuration



Configuring client-side RADIUS support

Each Brocade device client must be individually configured to use RADIUS servers. You use the **radius-server** command to specify the server IP address, authentication protocols, and other parameters. You can configure a maximum of 5 RADIUS servers on a Brocade device for AAA service.

NOTE
RADIUS requires that you configure both the client and the server.

The following table describes the parameters associated with a RADIUS server that is configured on the device.

TABLE 7 RADIUS server parameters

Parameter	Description
host	IP address (IPv4 or IPv6) or host name of the RADIUS server. Host name requires prior DNS configuration. The maximum supported length for the host name is 40 characters.
auth-port	The user datagram protocol (UDP) port used to connect the RADIUS server for authentication. The port range is 0 through 65535; the default port is 1812.
protocol	The authentication protocol to be used. Options include CHAP, PAP, and PEAP. The default protocol is CHAP. IPv6 hosts are not supported if PEAP is the configured protocol.
key	The shared secret between the device and the RADIUS server. The default value is "sharedsecret." The key cannot contain spaces and must be from 8 through 40 characters in length. Empty keys are not supported.
retries	The number of attempts permitted to connect to a RADIUS server. The range is 0 through 100, and the default value is 5.
timeout	Time to wait for a server to respond. The range is 1 through 60 seconds. The default value is 5 seconds.
encryption-level	Whether the encryption key should be stored in clear-text or in encrypted format. Default is 7 (encrypted). Possible values are 0 or 7, where 0 represents store the key in clear-text format and 7 represents encrypted format.

NOTE
If you do not configure the **key** attribute, the authentication session will not be encrypted. The value of the **key** attribute must match the value configured in the RADIUS configuration file; otherwise, the communication between the server and the device fails.

Refer also to:

- Adding a RADIUS server to the client server list on page 48
- Modifying the client-side RADIUS server configuration on page 49
- Configuring the client to use RADIUS for login authentication on page 50

Adding a RADIUS server to the client server list

You must configure the Domain Name System (DNS) server on the switch prior to adding the RADIUS server with a domain name or a host name. Without the DNS server, name resolution of the RADIUS server fails and therefore the add operation fails. Use the **ip dns** command to configure the DNS server.

NOTE
When a list of servers is configured on the switch, failover from one server to another server happens only if a RADIUS server fails to respond; it does not happen when user authentication fails.

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.

```
switch# configure terminal
Entering configuration mode terminal
```

2. Enter the **radius-server** command with the specified parameters.

```
switch(config)# radius-server host 10.38.37.180 protocol pap key "new#virgo*secret" timeout 10
```

   Once you run this command, you are placed into the AAA server configuration submode where you can specify additional parameters.

3. Enter the **exit** command to return to global configuration mode.

```
switch(config-host-10.38.37.180)# exit
```

4. Enter the **do show running-config radius-server host** *host_IP* command to verify the configuration.

```
switch# show running-config radius-server host 10.38.37.180
radius-server host 10.38.37.180
protocol  pap
key        "new# virgo*secret"
timeout    10
```

Modifying the client-side RADIUS server configuration

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.

```
switch# configure terminal
Entering configuration mode terminal
```

2. Enter the **radius-server host** command with the help option (**?**) to display the configured RADIUS servers.

```
switch(config)# radius-server ?
Possible completions:
<hostname: IP Address or Hostname of this RADIUS server>
10.38.37.180
10.24.65.6
```

3. Enter the **radius-server host** command with the IP address of the server you want to modify.

```
switch(config)# radius-server host 10.38.37.180
```

   After you run this command you are placed into the RADIUS server configuration submode where you can specify the parameters you want to modify.

4. Enter the parameters and values you want to change.

```
switch(config-host-10.38.37.180 )# key "changedsec"
switch(config-host-10.38.37.180 )# timeout 3
```

5. Enter the **do show running-config radius-server** command to verify the configuration.

   NOTE
   This command does not display default values.

```
switch(config)# do show running-config radius-server host 10.24.65.6
radius-server host 10.24.65.6
protocol  pap
key        changedsec
timeout    3
```

   NOTE
   The **no radius-server host** command removes the server configuration from the list of configured RADIUS servers. When used with a specified parameter, the command sets the default value of that parameter.

Configuring the client to use RADIUS for login authentication

After you configured the client-side RADIUS server list, you must set the authentication mode so that RADIUS is used as the primary source of authentication. Refer to Login authentication mode on page 29 for information on how to configure the login authentication mode.

RADIUS two factor authentication support

Traditional password-based authentication methods are based on "one-factor" authentication, where a user confirms an identity using a memorized password. Reliance on one-factor authentication exposes enterprises to increased security risks; passwords may be stolen, guessed, cracked, replayed, or compromised in other ways by unsolicited users by using Man in the Middle Attack.

Two factor authentication increases the security by adding an additional step to the basic log-in procedure which requires the user to have both the password and RSA Secure ID credentials from a hardware token before being able to access a device. The authentication proceeds as four basic steps:

First, each hardware token is assigned to a user. It generates an authentication code every 60 seconds using built-in clock and the card's random key (seed). This seed is 128 bits long, is different for each hardware-token, and is loaded into the RSA Secure ID server (RSA Authentication Manager). The token hardware is designed to be tamper-resistant to deter reverse engineering of the token. Network OS only supports an RSA ID key fob as a secondary authentication token.

Secondly, the RSA Authentication Manager authenticates the user's password or PIN and token's combination. It takes the clock time as the input value for the encryption process and it is encrypted with the seed record. The resulting value is the token.

Third, the RSA Agent receives authentication requests and forwards them to the RSA Authentication Manager through a secure channel. Based on the response from the Authentication Manager, agents either allow or deny user access.

Finally, the RSA RADIUS Server forwards the user's user ID and passes code to the RSA Authentication Manager, which verifies that the user ID exists and that the pass code is correct for that user at that specific time.

Each RSA Secure ID token holder must have a user record in the RSA Authentication Manager database. There are four ways to create these records:

- Adding data for individual users in the Add User dialog box.
- Copy and edit an existing user record to make a template with group membership and Agent Host activation lists that can be used for many new users.
- Import user data from Security Accounts Manager (SAM) database on a Windows NT system to the Authentication Manager using dumpsamusers.exe and loadsamusers.exe tools.
- Import user data from an LDAP directory.

The user records must be synchronized in order to operate. When synchronizing LDAP user records, the Database Administration application provides LDAP synchronization tools those can be used to populate the Authentication Manager's user database and keep it synchronized with LDAP directory server. The RSA Authentication Manager supports the following LDAP directory servers; Microsoft Active Directory, Sun ONE Directory Server, and Novell NDS eDirectory.

Using commands in Database Administration, you can add, edit, copy, list, delete, and run synchronization jobs to automatically maintain LDAP user records in the RSA Authentication Manager Database. Refer the RSA ACE/Server documentation for detailed info on adding the users and other configurations.

In order to support two factor authentication install RSA Authentication Manager on your Radius Server and set to it accept two-factor authentication input. When the user logs in, the password/tokencode works automatically without any changes to the Brocade switch, as shown in the following example.

```
        Welcome to Console Server Management Server HQ1-4E23-TS1 port S026

    HQ1-4E23-TS1 login: muser34
```

```
Password: ****************  <-----For example password/8675309

device#
```

TACACS+ Server Authentication

Understanding and configuring TACACS+

The Terminal Access Controller Access-Control System Plus (TACACS+) is an AAA server protocol that uses a centralized authentication server and multiple network access servers or clients. With TACACS+ support, management of Brocade devices seamlessly integrates into these environments. Once configured to use TACACS+, a Brocade device becomes a network access server.

If you are in logical chassis cluster mode, the configuration is applied to all nodes in the cluster.

TACACS+ authorization

The TACACS+ server is used only for authentication and accounting. Authorization is enforced by the Brocade role-based access control (RBAC) protocol at the device level. The same role should be assigned to a user configured on the TACACS+ server and configured on the device. If the device fails to get the user's role from the TACACS+ server after successful authentication, or if the role does not match any of the roles present on the device, the **user** role is assigned by default. Thereafter, the **brcd-role** is the key used to set the role from the TACACS+ server is used for RBAC.

TACACS+ authentication through management interfaces

You can access the device through the serial port, or through Telnet or SSH from either the management interface or the data ports (TE interface or in-band). The device goes through the same TACACS+-based authentication with either access method.

Supported TACACS+ packages and protocols

Brocade supports the following TACACS+ packages for running the TACACS+ daemon on remote AAA servers:

- Free TACACS+ daemon. You can download the latest package from www.shrubbery.net/tac_plus.
- ACS 5.3
- ACS 4.2

The TACACS+ protocol v1.78 is used for AAA services between the Brocade switch client and the TACACS+ server.

The authentication protocols supported for user authentication are Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP).

TACACS+ configuration components

Configuring TACACS+ requires configuring TACACS+ support on the client (including optional accounting), as well as configuring TACACS+ on the server. Support for mixed environments may also be required.

Configuring the client for TACACS+ support

Each Brocade device client must be individually configured to use TACACS+ servers. You use the **tacacs-server** command to specify the server IP address, authentication protocols, and other parameters. You can configure a maximum of five TACACS+ servers on a Brocade device for AAA service.

The parameters in the following table are associated with a TACACS+ server that is configured on the device.

TABLE 8 TACACS+ server parameters

Parameter	Description
host	IP address (IPv4 or IPv6) or domain/host name of the TACACS+ server. Host name requires prior DNS configuration. The maximum supported length for the host name is 40 characters.
port	The TCP port used to connect the TACACS+ server for authentication. The port range is 1 through 65535; the default port is 49.
protocol	The authentication protocol to be used. Options include CHAP and PAP. The default protocol is CHAP.
key	The shared secret between the device and the TACACS+ server. The default value is "sharedsecret." The key cannot contain spaces and must be from 8 through 40 characters in length. Empty keys are not supported.
retries	The number of attempts permitted to connect to a TACACS+ server. The range is 0 through 100, and the default value is 5.
timeout	The maximum amount of time to wait for a server to respond. Options are from 1 through 60 seconds, and the default value is 5 seconds.
encryption-level	Whether the encryption key should be stored in clear-text or in encrypted format. Default is 7 (encrypted). Possible values are 0 or 7, where 0 represents store the key in clear-text format and 7 represents encrypted format.

NOTE

If you do not configure the **key** attribute, the authentication session will not be encrypted. The value of **key** must match with the value configured in the TACACS+ configuration file; otherwise, the communication between the server and the device fails.

Refer also to:

- Adding a TACACS+ server to the client server list on page 54
- Modifying the client-side TACACS+ server configuration on page 55
- Configuring the client to use TACACS+ for login authentication on page 56
- Configuring TACACS+ accounting on the client side on page 56

Adding a TACACS+ server to the client server list

Prior to adding the TACACS+ server with a domain name or a host name, you must configure the Domain Name System (DNS) server on the device. Without the DNS server, the TACACS+ server name resolution fails and therefore the add operation fails. Use the **ip dns** command to configure the DNS server.

NOTE

When a list of servers is configured, failover from one server to another server happens only if a TACACS+ server fails to respond; it does not happen when user authentication fails.

The following procedure adds a TACACS+ server host in IPv6 format.

1. In the privileged EXEC mode, enter **configure terminal** to enter the global configuration mode.

```
switch# configure terminal
Entering configuration mode terminal
```

2. Enter **tacacs-server** and specify the server IP address.

```
switch(config)# tacacs-server host fec0:60:69bc:94:211:25ff:fec4:6010
```

   Upon execution of the command you are placed into the TACACS server configuration submode where you can specify additional parameters.

3. Specify the additional parameters.
   This example specifies the CHAP protocol key.

```
switch(config-tacacs-server-fec0:60:69bc:94:211:25ff:fec4:6010)# protocol chap key
"new#hercules*secret"
switch(config-tacacs-server-fec0:60:69bc:94:211:25ff:fec4:6010)# exit
switch(config)# do show running-config tacacs-server fec0:60:69bc:94:211:25ff:fec4:6010
tacacs-server host fec0:60:69bc:94:211:25ff:fec4:6010 key new# Hercules*secret
```

4. Enter **exit** to return to the global configuration mode.

```
switch(config-tacacs-server-fec0:60:69bc:94:211:25ff:fec4:6010)# exit
```

5. Enter **do show running-config tacacs-server host** *server_address* to verify the configuration.

```
switch(config)# do show running-config tacacs-server fec0:60:69bc:94:211:25ff:fec4:6010
tacacs-server host fec0:60:69bc:94:211:25ff:fec4:6010
key new# Hercules*secret
```

Modifying the client-side TACACS+ server configuration

1. In privileged EXEC mode, enter **configure terminal** to change to global configuration mode.

```
switch# configure terminal
Entering configuration mode terminal
```

2. Enter **tacacs-server host** with the help option (**?**) to display the configured server IP addresses.

```
switch(config)# tacacs-server host ?
fec0:60:69bc:94:211:25ff:fec4:6010
```

3. Enter **tacacs-server host** followed by the address of the server you wish to modify.

```
switch(config)# tacacs-server host fec0:60:69bc:94:211:25ff:fec4:6010
```

   Upon execution of the command you are placed into the TACACS server configuration submode where you can specify the parameters you want to modify.

4. Specify the additional parameters.

```
switch(config-tacacs-server-fec0:60:69bc:94:211:25ff:fec4:6010)# key "changedsec" retries 100
```

5. Enter **exit** to return to the global configuration mode.

```
switch(config-tacacs-server-fec0:60:69bc:94:211:25ff:fec4:6010)# exit
```

6. Enter **do show running-config tacacs-server** *server_address* to verify the configuration.

   This command does not display default values.

```
switch(config)# do show running-config tacacs-server fec0:60:69bc:94:211:25ff:fec4:6010
tacacs-server host fec0:60:69bc:94:211:25ff:fec4:6010
key       changedesc
retries   100!
```

   The **no tacacs-server host** command removes the server configuration from the list of configured RADIUS servers. If the TACACS server being deleted is the last one in the list and authentication mode is set to **tacacs+**, deletion of the server from the switch configuration is denied. When used with a specified parameter, the command sets the default value of that parameter.

Configuring the client to use TACACS+ for login authentication

After you configured the client-side TACACS+ server list, you must set the authentication mode so that TACACS+ is used as the primary source of authentication. refer to Login authentication mode on page 29 for information on how to configure the login authentication mode.

Configuring TACACS+ accounting on the client side

Once the fundamentals of TACACS+ authentication support are configured on the client, a variety of options are available for tracking user activity.

Client-side TACACS+ accounting overview

The TACACS+ protocol supports accounting as a function distinctly separate from authentication. You can use TACACS+ for authentication only, for accounting only, or for both. With a TACACS+ server you can track user logins and the commands users execute during a login session by enabling login accounting, command accounting, or both.

If you are in logical chassis cluster mode, the configuration is applied to all nodes in the cluster.

- When login accounting is enabled, the device sends a TACACS+ start accounting packet with relevant attributes to the configured TACACS+ server when the user logs in, and a stop accounting packet when the session terminates.
- When command accounting is enabled, the device sends a TACACS+ stop accounting packet to the server when the command execution completes. No TACACS+ start accounting packet is sent for command accounting. Most configuration commands, **show** commands and non-configuration commands such as **firmware download** will be tracked. Commands received through the NetConf interface will also be tracked.

If a TACACS+ server is used for both authentication and accounting, the device first attempts to connect to the TACACS+ server that was successfully used for authentication when sending accounting packets to the server. If the TACACS+ server cannot be reached, the device attempts to send the packets to the next server on the list. Note that there is no fail back in this case. When the first TACACS+ server becomes reachable again, the accounting packets continue to be sent to the second TACACS+ server.

If authentication is performed through some other mechanism, such as the device-local database, a RADIUS, or an LDAP server, the device will attempt to send the accounting packets to the first configured TACACS+ server. If that server is unreachable, the device will attempt to send the accounting packets to subsequent servers in the order in which they are configured.

Conditions for conformance

- Only login and command accounting is supported. System event accounting is not supported.

- You can use a TACACS+ server for accounting regardless of whether authentication is performed through RADIUS, LDAP, TACACS+, or the device-local user database. The only precondition is the presence of one or more TACACS+ servers configured on the device.
- No accounting can be performed if authentication fails.
- In command accounting, commands with partial timestamp cannot be logged. For example, a **firmware download** command issued with the **reboot** option will not be accounted for, because there is no timestamp available for completion of this command.

Firmware downgrade considerations

Before downgrading to a version that does not support TACACS+ accounting, you must disable both login and command accounting or the firmware download will fail with an appropriate error message.

Configuring TACACS+ accounting on the client

By default, accounting is disabled on the TACACS+ client (the device) and you must explicitly enable the feature. Enabling command accounting and login accounting on the TACACS+ client are two distinct operations. To enable login or command accounting, at least one TACACS+ server must be configured. Similarly, if either login or command accounting is enabled, if it is the only server in the list, you cannot remove a TACACS+ server.

Enabling login accounting

The following procedure enables login accounting on a switch where accounting is disabled.

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.

```
switch# configure terminal
Entering configuration mode terminal
```

2. Enter the **aaa accounting default exec start-stop tacacs+** command to enable login accounting.

```
switch(config)# aaa accounting exec default start-stop tacacs+
```

3. Enter **exit** to return to privileged EXEC mode.

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

4. Enter the **show running-config aaa accounting** command to verify the configuration.

```
switch(config)# show running-config aaa accounting
aaa accounting exec default start-stop tacacs+
aaa accounting commands default start-stop tacacs+
```

Enabling command accounting

The following procedure enables command accounting on a switch where login accounting is enabled and command accounting is disabled.

1. In privileged EXEC mode, enter **configure terminal** to enter global configuration mode.

```
switch# configure terminal
Entering configuration mode terminal
```

2. Enter **aaa accounting default command start-stop tacacs+** to enable command accounting.

```
switch(config)# aaa accounting command default start-stop tacacs+
```

3. Enter **exit** to return to privileged EXEC mode.

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

4. Enter **show running-config aaa accounting** to verify the configuration.

```
switch# show running-config aaa accounting
aaa accounting exec default start-stop none
aaa accounting commands default start-stop tacacs+
```

Disabling accounting

You have two options to disable accounting, either by using the **aaa accounting** command, with the none option or by using the **no** form of the command. Both variants are functionally equivalent. You must perform the disable operation separately for login accounting and for command accounting. The operation is performed in the global configuration mode.

This example shows two ways of disabling command accounting. The commands are executed in the global configuration mode.

```
switch(config)# aaa accounting commands default start-stop none
switch(config)# no aaa accounting commands default start-stop
```

This example shows two ways of disabling login accounting.

```
switch(config)# aaa accounting exec default start-stop none
switch(config)# no aaa accounting exec default start-stop
```

Viewing the TACACS+ accounting logs

The following excerpts from TACACS+ accounting logs exemplify typical success and failure cases for command and login accounting.

These examples were taken from the free TACACS+ server. The order of the attributes may vary depending on the server package, but the values are the same. The location of the accounting logs depend on the server configuration.

Command accounting examples

The following example shows a successful execution of the **shutdown** command by the admin user, followed by a **no shutdown** command.

```
Wed Oct 14 10:40:40 2015        10.18.245.157   admin1  /dev/pts/0      10.70.7.36      stop
task_id=1       timezone=Etc/GMT        service=shell   priv-lvl=0      Cmd="operational top configure
terminal" Stop_time=Wed Oct 14 17:39:49 2015

        Status=Succeeded

Wed Oct 14 10:42:14 2015        10.18.245.157   admin1  /dev/pts/0      10.70.7.36      stop
task_id=1       timezone=Etc/GMT        service=shell   priv-lvl=0      Cmd="configure conf-if-te-157/0/5
shutdown"      Stop_time=Wed Oct 14 17:41:24 2015

        Status=Succeeded

Wed Oct 14 10:42:23 2015        10.18.245.157   admin1  /dev/pts/0      10.70.7.36      stop
task_id=1       timezone=Etc/GMT        service=shell   priv-lvl=0      Cmd="configure conf-if-te-157/0/5
no shutdown"   Stop_time=Wed Oct 14 17:41:33 2015
```

The following example shows a successful execution of the **username** command by the admin user.

```
<102> 2012-04-09 15:21:43 4/9/2012 3:21:43 PM NAS_IP=10.17.37.150 Port=0 rem_addr=Console User=admin
Flags=Stop task_id=1 timezone=Etc/GMT+0 service=shell priv-lvl=0 Cmd=username Stop_time=Mon Apr 9 09:43:56
```

```
2012
 Status=Succeeded
```

The following example shows a failed execution of the **radius-server** command by the admin user due to an invalid host name or server IP address.

```
<102> 2012-04-09 14:19:42 4/9/2012 2:19:42 PM NAS_IP=10.17.37.150 Port=0 rem_addr=Console User=admin
Flags=Stop task_id=1 timezone=Etc/GMT+0 service=shell priv-lvl=0 Cmd=radius-server Stop_time=Mon Apr 9
08:41:56 2012
 Status=%% Error: Invalid host name or IP address
```

Login (EXEC) accounting examples

The following example shows a successful login of the trial user.

```
<102> 2012-05-14 11:47:49 5/14/2012 11:47:49 AM NAS_IP=10.17.46.42 Port=/dev/ttyS0 rem_addr=Console
User=trial Flags=Start task_id=1 timezone=Asia/Kolkata service=shell
```

The following example shows a successful logout of the trial user.

```
<102>2012-05-14 11:49:52 5/14/2012 11:49:52 AM NAS_IP=10.17.46.42 Port=/dev/ttyS0 rem_addr=console
User=trial Flags=Stop task_id=1 timezone=Asia/Kolkata service=shell elapsed_time=123 reason=admin reset
```

Configuring TACACS+ on the server side

Step-by-step instructions for installing and configuring can be obtained from www.cisco.com. Confer with your system or network administrator prior to configuration for any special needs your network environment may have.

Server-side user account administration overview

With TACACS+ servers, you should set up user accounts by their true network-wide identity, rather than by the account names created on a Brocade device. Along with each account name, you must assign appropriate device access roles. A user account can exist on TACACS+ server with the same name as a user on the device at the same time.

When logging in to a device configured with a TACACS+ server, users enter their assigned TACACS+ account names and passwords when prompted. Once the TACACS+ server authenticates a user, it responds with the assigned device role and information associated with the user account information using a Brocade Vendor-Specific Attribute (VSA). An Authentication-Accept response without the role assignment automatically grants the "user" role.

User accounts, protocols passwords, and related settings are configured by editing the server configuration files. The following configuration examples are based on the documentation provided by Cisco for its TACACS+ daemon users.

Establishing a server-side user account

The following example assigns the user "Mary" the Brocade role of "vlanadmin" and different passwords depending on whether the CHAP or the PAP protocol is used. In the following example, the brcd-role attribute is mandatory, which works in a Brocade-only environment. In a mixed vendor environment, the brcd-role attribute most be set to optional. Refer to Configuring TACACS+ for a mixed vendor environment on page 61 for more information.

```
user = Mary {
chap = cleartext "chap password"
pap = cleartext "pap password"
service = exec {
brcd-role = vlanadmin;
}
}
```

The following example assigns the user "Agnes" a single password for all types of login authentication.

```
user = Agnes {
global = cleartext "Agnes global password"
}
```

Alternatively, a user can be authenticated using the /etc/passwd file. Configure the account as shown in the following example.

```
user = fred {
login = file /etc/passwd
}
```

Changing a server-side TACACS+ account password

Changing a TACACS+ user password is done on the server by editing the TACACS+ server configuration file.

Defining a server-side TACACS+ group

A TACACS+ group or role can contain the same attributes as the users. By inference, all the attributes of a group can be assigned to any user to whom the group is assigned. The TACACS+ group, while functionally similar to the Brocade role concept, has no relation with the value of "brcd-role" attribute.

The following example defines a TACACS+ group.

```
group = admin {
# group admin has a cleartext password which all members share
# unless they have their own password defined
chap = cleartext "my$parent$chap$password"
}
```

The following example assigns the user "Brocade" with the group "admin".

```
user = Brocade {
member = admin
pap = cleartext "pap password"
}
```

Setting a server-side account expiration date

You can set an expiration date for an account by using the "expires" attribute in the TACACS+ server configuration file. The expiration date has the format "*MMM DD YYYY* "

```
user = Brocade {
member = admin
expires = "Jan 1 2011"
pap = cleartext "pap password"
}
```

Configuring a TACACS+ server key

The TACACS+ server key is the shared secret used to secure the messages exchanged between the Brocade switch and the TACACS+ server. The TACACS+ server key must be configured on both the TACACS+ server and the client Brocade switch. Only one key is defined per server in the TACACS+ server configuration file. The key is defined as follows:

```
key = "vcs shared secret"
```

Configuring TACACS+ for a mixed vendor environment

Network OS uses Role Based Access Control (RBAC) to authorize access to system objects by authenticated users. In AAA environments users may need to be authorized across Brocade and non-Brocade platforms. You can use TACACS+ to provide centralized AAA services to multiple network access servers or clients. To use TACACS+ services in multi-vendor environments, you must configure the Attribute-Value Pair (AVP) argument to be optional as shown in the example.

```
brcd-role*admin
```

The Network OS device sends the optional argument 'brcd-role' in the authorization request to the TACACS+ service. Most TACACS+ servers are programmed to return the same argument in response to the authorization request, If 'brcd-role' is configured as an optional argument, it is sent in the authorization request and Network OS users are able to successfully authorize with all TACACS+ services in a mixed-vendor environment.

Example: Configuring optional arguments in tac_plus

The following is a specific example for tac_plus package. Syntax for other packages may differ.

In the example, the mandatory attribute priv-lvl=15 is set to allow Cisco to authenticate. The optional brcd-role = admin argument is added to the tac_plus.conf file and allows Brocade VDX switches to authenticate.

The following example configures a user with the optional attribute value pair, brcd-role = admin. A Brocade user must match both the *username* and *usergroup* to authenticate successfully.

```
user = <username> {
    default service = permit
    service = exec {
     priv-lvl=15
  optional brcd-role = admin
  }
}
```

or

```
group = <usergroup> {
    default service = permit
    service = exec {
     priv-lvl=15
     optional brcd-role = admin
  }
}
user = <username> {
   Member = <usergroup>
}
```

TACACS and TACACS+ security

You can use the security protocol Terminal Access Controller Access Control System (TACACS) or TACACS+ to authenticate the following kinds of access to the Brocade device:

- Telnet access
- SSH access
- Console access
- Web management access
- Access to the Privileged EXEC level and CONFIG levels of the CLI

The TACACS and TACACS+ protocols define how authentication, authorization, and accounting information is sent between a Brocade device and an authentication database on a TACACS/TACACS+ server. TACACS/TACACS+ services are maintained in a database, typically on a UNIX workstation or PC with a TACACS/TACACS+ server running.

How TACACS+ differs from TACACS

TACACS is a simple UDP-based access control protocol originally developed by BBN for MILNET. TACACS+ is an enhancement to TACACS and uses TCP to ensure reliable delivery.

TACACS+ is an enhancement to the TACACS security protocol. TACACS+ improves on TACACS by separating the functions of authentication, authorization, and accounting (AAA) and by encrypting all traffic between the Brocade device and the TACACS+ server. TACACS+ allows for arbitrary length and content authentication exchanges, which allow any authentication mechanism to be utilized with the Brocade device. TACACS+ is extensible to provide for site customization and future development features. The protocol allows the Brocade device to request very precise access control and allows the TACACS+ server to respond to each component of that request.

NOTE

TACACS+ provides for authentication, authorization, and accounting, but an implementation or configuration is not required to employ all three.

TACACS/TACACS+ authentication, authorization,and accounting

When you configure a Brocade device to use a TACACS/TACACS+ server for authentication , the device prompts users who are trying to access the CLI for a user name and password, then verifies the password with the TACACS/TACACS+ server.

If you are using TACACS+, Brocade recommends that you also configure authorization , in which the Brocade device consults a TACACS + server to determine which management privilege level (and which associated set of commands) an authenticated user is allowed to use. You can also optionally configure accounting , which causes the Brocade device to log information on the TACACS+ server when specified events occur on the device.

> NOTE
> By default, a user logging into the device from Telnet or SSH would first enter the User EXEC level. The user can enter the **enable** command to get to the Privileged EXEC level. A user that is successfully authenticated can be automatically placed at the Privileged EXEC level after login. Refer to Entering privileged EXEC mode after a Telnet or SSH login on page 73.

Configuring TACACS/TACACS+ for devices in a Brocade traditional stack

Because devices operating in a Brocade traditional stack topology present multiple console ports, you must take additional steps to secure these ports when configuring TACACS/TACACS+.

The following is a sample AAA console configuration using TACACS+.

```
aaa authentication login default tacacs+ enable
aaa authentication login privilege-mode
aaa authorization commands 0 default  tacacs+
aaa authorization exec default  tacacs+
aaa accounting commands 0 default start-stop  tacacs+
aaa accounting exec default start-stop  tacacs+
aaa accounting system default start-stop  tacacs+
enable aaa console
hostname Fred
ip address 10.10.6.56/255
tacacs-server host 255.253.255
tacacs-server key 2 $d3NpZ0BVXFpJ
```

kill console

Syntax: kill console [**all** | **unit**]

- **all** - logs out all console port on stack units that are not the Active Controller
- **unit** - logs out the console port on a specified unit

Once AAA console is enabled, you should log out any open console ports on your traditional stack using the **kill console** command:

```
device(config)#kill console all
```

In case a user forgets to log out or a console is left unattended, you can also configure the console timeout (in minutes) for active stack units.

```
device(config)#stack unit 1
device(config-unit-1)#console timeout 5
```

> NOTE
> Console timeout will not work for non-active and member units, use “kill console” command to log out all console sessions for non-active and member units. The sessions will re-authenticate for console access.

Use the **show who** and the **show telnet** commands to confirm the status of console sessions.

```
stack9#show who
Console connections (by unit number):
 1      established
        you are connecting to this session
        4 seconds in idle
 2      established
        1 hours 3 minutes 12 seconds in idle
 3      established
        1 hours 3 minutes 9 seconds in idle
 4      established
        1 hours 3 minutes 3 seconds in idle
Telnet connections (inbound):
 1      closed
 2      closed
 3      closed
 4      closed
 5      closed
Telnet connection (outbound):
 6      closed
SSH connections:
 1      closed
 2      closed
 3      closed
 4      closed
 5      closed
stack9#
stack9#show telnet
Console connections (by unit number):
 1      established
        you are connecting to this session
        1 minutes 5 seconds in idle
 2      established
        1 hours 4 minutes 18 seconds in idle
 3      established
        1 hours 4 minutes 15 seconds in idle
 4      established
        1 hours 4 minutes 9 seconds in idle
Telnet connections (inbound):
 1      closed
 2      closed
 3      closed
 4      closed
 5      closed
Telnet connection (outbound):
 6      closed
SSH connections:
 1      closed
 2      closed
 3      closed
 4      closed
 5      closed
stack9#
```

TACACS authentication

NOTE

Also, multiple challenges are supported for TACACS+ login authentication.

When TACACS authentication takes place, the following events occur.

1. A user attempts to gain access to the Brocade device by doing one of the following:
   - Logging into the device using Telnet, SSH, or the Web Management Interface
   - Entering the Privileged EXEC level or CONFIG level of the CLI

2. The user is prompted for a username and password.
3. The user enters a username and password.
4. The Brocade device sends a request containing the username and password to the TACACS server.
5. The username and password are validated in the TACACS server database.
6. If the password is valid, the user is authenticated.

TACACS+ authentication

When TACACS+ authentication takes place, the following events occur.

1. A user attempts to gain access to the Brocade device by doing one of the following:
   - Logging into the device using Telnet, SSH, or the Web Management Interface
   - Entering the Privileged EXEC level or CONFIG level of the CLI
2. The user is prompted for a username.
3. The user enters a username.
4. The Brocade device obtains a password prompt from a TACACS+ server.
5. The user is prompted for a password.
6. The user enters a password.
7. The Brocade device sends the password to the TACACS+ server.
8. The password is validated in the TACACS+ server database.
9. If the password is valid, the user is authenticated.

TACACS+ authorization

Brocade devices support two kinds of TACACS+ authorization:

- Exec authorization determines a user privilege level when they are authenticated
- Command authorization consults a TACACS+ server to get authorization for commands entered by the user

When TACACS+ exec authorization takes place, the following events occur.

1. A user logs into the Brocade device using Telnet, SSH, or the Web Management Interface
2. The user is authenticated.
3. The Brocade device consults the TACACS+ server to determine the privilege level of the user.
4. The TACACS+ server sends back a response containing an A-V (Attribute-Value) pair with the privilege level of the user.
5. The user is granted the specified privilege level.

When TACACS+ command authorization takes place, the following events occur.

1. A Telnet, SSH, or Web Management Interface user previously authenticated by a TACACS+server enters a command on the Brocade device.
2. A Telnet, SSH, or Web Management Interface user previously authenticated by a TACACS+server enters a command on the Brocade device.
3. The Brocade device looks at its configuration to see if the command is at a privilege level that requires TACACS+ command authorization.

4. If the command belongs to a privilege level that requires authorization, the Brocade device consults the TACACS+ server to see if the user is authorized to use the command.
5. If the user is authorized to use the command, the command is executed.

TACACS+ accounting

TACACS+ accounting works as follows.

1. One of the following events occur on the Brocade device:
   - A user logs into the management interface using Telnet or SSH
   - A user enters a command for which accounting has been configured
   - A system event occurs, such as a reboot or reloading of the configuration file
2. The Brocade device checks the configuration to see if the event is one for which TACACS+ accounting is required.
3. If the event requires TACACS+ accounting, the Brocade device sends a TACACS+ Accounting Start packet to the TACACS+ accounting server, containing information about the event.
4. The TACACS+ accounting server acknowledges the Accounting Start packet.
5. The TACACS+ accounting server records information about the event.
6. When the event is concluded, the Brocade device sends an Accounting Stop packet to the TACACS+ accounting server.
7. The TACACS+ accounting server acknowledges the Accounting Stop packet.

AAA operations for TACACS/TACACS+

The following table lists the sequence of authentication, authorization, and accounting operations that take place when a user gains access to a Brocade device that has TACACS/TACACS+ security configured.

User action	Applicable AAA operations
User attempts to gain access to the Privileged EXEC and CONFIG levels of the CLI	Enable authentication: aaa authentication enable default method-list
	Exec authorization (TACACS+): aaa authorization exec default tacacs+
	System accounting start (TACACS+): aaa accounting system default start-stop method-list
User logs in using Telnet/SSH	Login authentication: aaa authentication login default method-list
	Exec authorization (TACACS+): aaa authorization exec default tacacs+
	Exec accounting start (TACACS+): aaa accounting exec default method-list
	System accounting start (TACACS+): aaa accounting system default start-stop method-list
User logs into the Web Management Interface	Web authentication: aaa authentication web-server default <method-list>
	Exec authorization (TACACS+): aaa authorization exec default tacacs+
User logs out of Telnet/SSH session	Command accounting (TACACS+): aaa accounting commands privilege-level default start-stop method-list
	EXEC accounting stop (TACACS+): aaa accounting exec default start-stop method-list
User enters system commands (for example, reload , boot system)	Command authorization (TACACS+): aaa authorization commands privilege-level default method-list
	Command accounting (TACACS+): aaa accounting commands privilege-level default start-stop method-list

User action	Applicable AAA operations
	System accounting stop (TACACS+): aaa accounting system default start-stop method-list
User enters the command: [no] aaa accounting system defaultstart-stop method-list	Command authorization (TACACS+): aaa authorization commands privilege-level default method-list
	Command accounting (TACACS+): aaa accounting commands privilege-level default start-stop method-list
	System accounting start (TACACS+): aaa accounting system default start-stop method-list

AAA security for commands pasted into the running-config

If AAA security is enabled on the device, commands pasted into the running-config are subject to the same AAA operations as if they were entered manually.

When you paste commands into the running-config, and AAA command authorization or accounting, or both, are configured on the device, AAA operations are performed on the pasted commands. The AAA operations are performed before the commands are actually added to the running-config. The server performing the AAA operations should be reachable when you paste the commands into the running-config file. If the device determines that a pasted command is invalid, AAA operations are halted on the remaining commands. The remaining commands may not be executed if command authorization is configured.

TACACS/TACACS+ configuration considerations

- You must deploy at least one TACACS/TACACS+ server in your network.
- Brocade devices support authentication using up to eight TACACS/TACACS+ servers. The device tries to use the servers in the order you add them to the device configuration.
- You can select only one primary authentication method for each type of access to a device (CLI through Telnet, CLI Privileged EXEC and CONFIG levels). For example, you can select TACACS+ as the primary authentication method for Telnet CLI access, but you cannot also select RADIUS authentication as a primary method for the same type of access. However, you can configure backup authentication methods for each access type.
- You can configure the Brocade device to authenticate using a TACACS or TACACS+ server, not both.

Configuring TACACS

Follow the procedure given below for TACACS configurations.

1. Identify TACACS servers. Refer to Identifying the TACACS/TACACS+ servers on page 69.
2. Set optional parameters. Refer to Setting optional TACACS and TACACS+ parameters on page 70.
3. Configure authentication-method lists. Refer to Configuring authentication-method lists forTACACS and TACACS+ on page 72.

Configuring TACACS+

Follow the procedure given below for TACACS+ configurations.

1. Identify TACACS+ servers. Refer to Identifying the TACACS/TACACS+ servers on page 69.
2. Set optional parameters. Refer to Setting optional TACACS and TACACS+ parameters on page 70.

3. Configure authentication-method lists. Refer to Configuring authentication-method lists forTACACS and TACACS+ on page 72.
4. Optionally configure TACACS+ authorization. Refer to Configuring TACACS+ authorization on page 74.
5. Optionally configure TACACS+ accounting. Refer to TACACS+ accounting configuration on page 76.

Enabling TACACS

TACACS is disabled by default. To configure TACACS/TACACS+ authentication parameters, you must enable TACACS by entering the following command.

```
device(config)#enable snmp config-tacacs
```

Syntax: [no] enable snmp [**config-radius** | **config-tacacs**]

The config-radius parameter specifies the RADIUS configuration mode. RADIUS is disabled by default.

The config-tacacs parameter specifies the TACACS configuration mode. TACACS is disabled by default.

Identifying the TACACS/TACACS+ servers

To use TACACS/TACACS+ servers to authenticate access to a Brocade device, you must identify the servers to the Brocade device.

For example, to identify three TACACS/TACACS+ servers, enter commands such as the following.

```
device(config)#tacacs-server host 10.94.6.161
device(config)#tacacs-server host 10.94.6.191
device(config)#tacacs-server host 10.94.6.122
```

Syntax: tacacs-server host { *ip-addr* |*ipv6-addr* | *server-name* } [**auth-port** *number*] [**acct-port**number]

The ip-addr | ipv6-addr | hostname parameter specifies the IP address or host name of the server. You can enter up to eight **tacacs-server host** commands to specify up to eight different servers.

> NOTE
> To specify the server's host name instead of its IP address, you must first identify a DNS server using the **ip dns server-address** ip-addr command at the global CONFIG level.

If you add multiple TACACS/TACACS+ authentication servers to the Brocade device, the device tries to reach them in the order you add them. For example, if you add three servers in the following order, the software tries the servers in the same order.

1. 10.94.6.161
2. 10.94.6.191

3. 10.94.6.122

You can remove a TACACS/TACACS+ server by entering **no** followed by the **tacacs-server** command. For example, to remove 10.94.6.161, enter the following command.

```
device(config)#no tacacs-server host 10.94.6.161
```

NOTE

If you erase a **tacacs-server** command (by entering "no" followed by the command), make sure you also erase the **aaa** commands that specify TACACS/TACACS+ as an authentication method. (Refer to Configuring authentication-method lists forTACACS and TACACS+ on page 72.) Otherwise, when you exit from the CONFIG mode or from a Telnet session, the system continues to believe it is TACACS/TACACS+ enabled and you will not be able to access the system.

The **auth-port** parameter specifies the UDP (for TACACS) or TCP (for TACACS+) port number of the authentication port on the server. The default port number is 49.

Specifying different servers for individual AAA functions

In a TACACS+ configuration, you can designate a server to handle a specific AAA task. For example, you can designate one TACACS+ server to handle authorization and another TACACS+ server to handle accounting. You can set the TACACS+ key for each server.

To specify different TACACS+ servers for authentication, authorization, and accounting, enter the command such as following.

```
device(config)#tacacs-server host 10.2.3.4 auth-port 49 authentication-only key abc
device(config)#tacacs-server host 10.2.3.5 auth-port 49 authorization-only key def
device(config)#tacacs-server host 10.2.3.6 auth-port 49 accounting-only key ghi
```

Syntax: tacacs-server host { *ip-addr* | *ipv6-addr* | *server-name* } [**auth-port** *num*] [**authentication-only** | **authorization-only** | **accounting-only** | **default**] [**key** [**0** | **1**] *string*]

The default parameter causes the server to be used for all AAA functions.

After authentication takes place, the server that performed the authentication is used for authorization and accounting. If the authenticating server cannot perform the requested function, then the next server in the configured list of servers is tried; this process repeats until a server that can perform the requested function is found, or every server in the configured list has been tried.

Setting optional TACACS and TACACS+ parameters

You can set the following optional parameters in a TACACS and TACACS+ configuration:

- TACACS+ key - This parameter specifies the value that the Brocade device sends to the TACACS+ server when trying to authenticate user access.
- Retransmit interval - This parameter specifies how many times the Brocade device will resend an authentication request when the TACACS/TACACS+ server does not respond. The retransmit value can be from 1 - 5 times. The default is 3 times.
- Dead time - This parameter specifies how long the Brocade device waits for the primary authentication server to reply before deciding the server is dead and trying to authenticate using the next server. The dead-time value can be from 1 - 5 seconds. The default is 3 seconds.
- Timeout - This parameter specifies how many seconds the Brocade device waits for a response from a TACACS/TACACS+ server before either retrying the authentication request, or determining that the TACACS/TACACS+ servers are unavailable and

moving on to the next authentication method in the authentication-method list. The timeout can be from 1 - 15 seconds. The default is 3 seconds.

Setting the TACACS+ key

The **key** parameter in the **tacacs-server** command is used to encrypt TACACS+ packets before they are sent over the network. The value for the **key** parameter on the Brocade device should match the one configured on the TACACS+ server. The key can be from 1 - 32 characters in length and cannot include any space characters.

NOTE
The **tacacs-server key** command applies only to TACACS+ servers, not to TACACS servers. If you are configuring TACACS, do not configure a key on the TACACS server and do not enter a key on the Brocade device.

To specify a TACACS+ server key, enter a command such as following.

```
device(config)#tacacs-server key rkwong
```

Syntax: tacacs-server key [**0**] *string*

When you display the configuration of the Brocade device, the TACACS+ keys are encrypted. For example.

```
device(config)#
tacacs-server key abc
device(config)#write terminal
...
tacacs-server host 10.2.3.5 auth-port 49
tacacs key 2$!2d
```

NOTE
Encryption of the TACACS+ keys is done by default. The **0** parameter disables encryption. The **1** parameter is not required; it is provided for backwards compatibility.

Setting the retransmission limit

The **retransmit** parameter specifies how many times the Brocade device will resend an authentication request when the TACACS/TACACS+ server does not respond. The retransmit limit can be from 1 - 5 times. The default is 3 times.

To set the TACACS and TACACS+ retransmit limit, enter a command such as the following.

```
device(config)#tacacs-server retransmit 5
```

Syntax: tacacs-server retransmit *number*

Setting the timeout parameter

The **timeout** parameter specifies how many seconds the Brocade device waits for a response from the TACACS/TACACS+ server before either retrying the authentication request, or determining that the TACACS/TACACS+ server is unavailable and moving on to the next authentication method in the authentication-method list. The timeout can be from 1 - 15 seconds. The default is 3 seconds.

```
device(config)#tacacs-server timeout 5
```

Syntax: tacacs-server timeout *number*

Configuring authentication-method lists forTACACS and TACACS+

You can use TACACS/TACACS+ to authenticate Telnet/SSH access and access to Privileged EXEC level and CONFIG levels of the CLI. When configuring TACACS/TACACS+ authentication, you create authentication-method lists specifically for these access methods, specifying TACACS/TACACS+ as the primary authentication method.

Within the authentication-method list, TACACS/TACACS+ is specified as the primary authentication method and up to six backup authentication methods are specified as alternates. If TACACS/TACACS+ authentication fails due to an error, the device tries the backup authentication methods in the order they appear in the list.

When you configure authentication-method lists for TACACS/TACACS+ authentication, you must create a separate authentication-method list for Telnet/SSH CLI access, and for access to the Privileged EXEC level and CONFIG levels of the CLI.

To create an authentication method list that specifies TACACS/TACACS+ as the primary authentication method for securing Telnet/SSH access to the CLI.

```
device(config)#enable telnet authentication
device(config)#aaa authentication login default tacacs local
```

The commands above cause TACACS/TACACS+ to be the primary authentication method for securing Telnet/SSH access to the CLI. If TACACS/TACACS+ authentication fails due to an error with the server, authentication is performed using local user accounts instead.

To create an authentication-method list that specifies TACACS/TACACS+ as the primary authentication method for securing access to Privileged EXEC level and CONFIG levels of the CLI.

```
device(config)#aaa authentication enable default tacacs local none
```

The command above causes TACACS/TACACS+ to be the primary authentication method for securing access to Privileged EXEC level and CONFIG levels of the CLI. If TACACS/TACACS+ authentication fails due to an error with the server, local authentication is used instead. If local authentication fails, no authentication is used; the device automatically permits access.

The **web-server** | **enable** | **login** parameter specifies the type of access this authentication-method list controls. You can configure one authentication-method list for each type of access.

> NOTE
> If you configure authentication for Web management access, authentication is performed each time a page is requested from the server. When frames are enabled on the Web Management Interface, the browser sends an HTTP request for each frame. The Brocade device authenticates each HTTP request from the browser. To limit authentications to one per page, disable frames on the Web Management Interface.

The *method1* parameter specifies the primary authentication method. The remaining optional *method* parameters specify additional methods to try if an error occurs with the primary method. A method can be one of the values listed in the Method Parameter column in the following table.

TABLE 9 Authentication method values

Method parameter	Description
line	Authenticate using the password you configured for Telnet access. The Telnet password is configured using the **enable telnet password...** command. Refer to the Setting a telnet password task.
enable	Authenticate using the password you configured for the Super User privilege level. This password is configured using the **enable super-user-password...** command.

TABLE 9 Authentication method values (continued)

Method parameter	Description
	Refer to the Setting passwords for management privilege levels task.
local	Authenticate using a local user name and password you configured on the device. Local user names and passwords are configured using the **username...** command. Refer to the Local user account configuration task.
tacacs	Authenticate using the database on a TACACS server. You also must identify the server to the device using the **tacacs-server** command.
tacacs+	Authenticate using the database on a TACACS+ server. You also must identify the server to the device using the **tacacs-server** command.
radius	Authenticate using the database on a RADIUS server. You also must identify the server to the device using the **radius-server** command.
none	Do not use any authentication method. The device automatically permits access.

Entering privileged EXEC mode after a Telnet or SSH login

By default, a user enters User EXEC mode after a successful login through Telnet or SSH. Optionally, you can configure the device so that a user enters Privileged EXEC mode after a Telnet or SSH login. To do this, use the following command.

```
device(config)#aaa authentication login privilege-mode
```

Syntax: aaa authentication login privilege-mode

The user privilege level is based on the privilege level granted during login.

Configuring enable authentication to prompt for password only

If Enable authentication is configured on the device, when a user attempts to gain Super User access to the Privileged EXEC and CONFIG levels of the CLI, by default he or she is prompted for a username and password. You can configure the Brocade device to prompt only for a password. The device uses the username entered at login, if one is available. If no username was entered at login, the device prompts for both username and password.

To configure the Brocade device to prompt only for a password when a user attempts to gain Super User access to the Privileged EXEC and CONFIG levels of the CLI.

```
device(config)#aaa authentication enable implicit-user
```

Syntax: [no] aaa authentication enable implicit-user

Telnet and SSH prompts when the TACACS+ Server is unavailable

When TACACS+ is the first method in the authentication method list, the device displays the login prompt received from the TACACS+ server. If a user attempts to login through Telnet or SSH, but none of the configured TACACS+ servers are available, the following takes place:

- If the next method in the authentication method list is "enable", the login prompt is skipped, and the user is prompted for the Enable password (that is, the password configured with the **enable super-user-password** command).
- If the next method in the authentication method list is "line", the login prompt is skipped, and the user is prompted for the Line password (that is, the password configured with the **enable telnet password** command).

Configuring TACACS+ authorization

Brocade devices support TACACS+ authorization for controlling access to management functions in the CLI. Two kinds of TACACS+ authorization are supported:

- Exec authorization determines a user privilege level when they are authenticated
- Command authorization consults a TACACS+ server to get authorization for commands entered by the user

Configuring exec authorization

When TACACS+ exec authorization is performed, the Brocade device consults a TACACS+ server to determine the privilege level of the authenticated user. To configure TACACS+ exec authorization on the Brocade device, enter the following command.

```
device(config)#aaa authorization exec default tacacs+
```

Syntax: aaa authorization exec default tacacs+[**none**]

If you specify **none** , or omit the **aaa authorization exec** command from the device configuration, no exec authorization is performed.

A user privilege level is obtained from the TACACS+ server in the "foundry-privlvl" A-V pair. If the **aaa authorization exec default tacacs+** command exists in the configuration, the device assigns the user the privilege level specified by this A-V pair. If the command does not exist in the configuration, then the value in the "foundry-privlvl" A-V pair is ignored, and the user is granted Super User access.

> NOTE
> If the **aaa authorization exec default tacacs+** command exists in the configuration, following successful authentication the device assigns the user the privilege level specified by the "foundry-privlvl" A-V pair received from the TACACS+ server. If the **aaa authorization exec default tacacs+** command does not exist in the configuration, then the value in the "foundry-privlvl" A-V pair is ignored, and the user is granted Super User access.Also note that in order for the **aaa authorization exec default tacacs+** command to work, either the**aaa authentication enable default tacacs+** command, or the **aaa authentication login privilege-mode** command must also exist in the configuration.

Configuring an Attribute-Value pair on the TACACS+ server

During TACACS+ exec authorization, the Brocade device expects the TACACS+ server to send a response containing an A-V (Attribute-Value) pair that specifies the privilege level of the user. When the Brocade device receives the response, it extracts an A-V pair configured for the Exec service and uses it to determine the user privilege level.

To set a user privilege level, you can configure the "foundry-privlvl" A-V pair for the Exec service on the TACACS+ server.

```
user=bob {
   default service = permit
   member admin
   #Global password
   global = cleartext "cat"
   service = exec {
     foundry-privlvl = 0
        }
}
```

In this example, the A-V pair `foundry-privlvl = 0` grants the user full read-write access. The value in the foundry-privlvl A-V pair is an integer that indicates the privilege level of the user. Possible values are 0 for super-user level, 4 for port-config level, or 5 for read-only level. If a value other than 0, 4, or 5 is specified in the foundry-privlvl A-V pair, the default privilege level of 5 (read-only) is used. The foundry-privlvl A-V pair can also be embedded in the group configuration for the user. See your TACACS+ documentation for the configuration syntax relevant to your server.

If the foundry-privlvl A-V pair is not present, the Brocade device extracts the last A-V pair configured for the Exec service that has a numeric value. The Brocade device uses this A-V pair to determine the user privilege level.

```
user=bob {
   default service = permit
   member admin
   #Global password
   global = cleartext "cat"
   service = exec {
     privlvl = 15
        }
}
```

The attribute name in the A-V pair is not significant; the Brocade device uses the last one that has a numeric value. However, the Brocade device interprets the value for a non-"foundry-privlvl" A-V pair differently than it does for a "foundry-privlvl" A-V pair. The following table lists how the Brocade device associates a value from a non-"foundry-privlvl" A-V pair with a Brocade privilege level.

TABLE 10 Brocade equivalents for non-"foundry-privlvl" A-V pair values

Value for non-"foundry-privlvl" A-V pair	Brocade privilege level
15	0 (super-user)
From 14 - 1	4 (port-config)
Any other number or 0	5 (read-only)

In the example above, the A-V pair configured for the Exec service is `privlvl = 15` . The Brocade device uses the value in this A-V pair to set the user privilege level to 0 (super-user), granting the user full read-write access.

In a configuration that has both a "foundry-privlvl" A-V pair and a non-"foundry-privlvl" A-V pair for the Exec service, the non-"foundry-privlvl" A-V pair is ignored.

```
user=bob {
   default service = permit
   member admin
   #Global password
   global = cleartext "cat"
   service = exec {
     foundry-privlvl = 4
     privlvl = 15
        }
}
```

In this example, the user would be granted a privilege level of 4 (port-config level). The `privlvl = 15` A-V pair is ignored by the Brocade device.

If the TACACS+ server has no A-V pair configured for the Exec service, the default privilege level of 5 (read-only) is used.

Configuring command authorization

When TACACS+ command authorization is enabled, the Brocade device consults a TACACS+ server to get authorization for commands entered by the user.

You enable TACACS+ command authorization by specifying a privilege level whose commands require authorization. For example, to configure the Brocade device to perform authorization for the commands available at the Super User privilege level (that is, all commands on the device), enter the following command.

```
device(config)#aaa authorization commands 0 default tacacs+
```

Syntax: aaa authorization commands *privilege-level* **default** [**tacacs+** | **radius** | **none**]

The privilege-level parameter can be one of the following:

- 0 - Authorization is performed for commands available at the Super User level (all commands)
- 4 - Authorization is performed for commands available at the Port Configuration level (port-config and read-only commands)
- 5 - Authorization is performed for commands available at the Read Only level (read-only commands)

NOTE
TACACS+ command authorization can be performed only for commands entered from Telnet or SSH sessions, or from the console. No authorization is performed for commands entered at the Web Management Interface.

TACACS+ command authorization is not performed for the following commands:

- At all levels: **exit** , **logout** , **end** , and **quit** .
- At the Privileged EXEC level: **enable** or **enable** text , where text is the password configured for the Super User privilege level.

If configured, command accounting is performed for these commands.

AAA support for console commands

AAA support for commands entered at the console includes the following:

- Login prompt that uses AAA authentication, using authentication-method Lists
- Exec Authorization
- Exec Accounting
- Command authorization
- Command accounting
- System Accounting

To enable AAA support for commands entered at the console, enter the following command.

```
device(config)#enable aaa console
```

Syntax: [no] enable aaa console

TACACS+ accounting configuration

Brocade devices support TACACS+ accounting for recording information about user activity and system events. When you configure TACACS+ accounting on a Brocade device, information is sent to a TACACS+ accounting server when specified events occur, such as when a user logs into the device or the system is rebooted.

Configuring TACACS+ accounting for Telnet/SSH (Shell) access

To send an Accounting Start packet to the TACACS+ accounting server when an authenticated user establishes a Telnet or SSH session on the Brocade device, and an Accounting Stop packet when the user logs out.

```
device(config)#aaa accounting exec default start-stop tacacs+
```

Syntax: aaa accounting exec default start-stop [**tacacs+** | **radius** | **none**]

Configuring TACACS+ accounting for CLI commands

You can configure TACACS+ accounting for CLI commands by specifying a privilege level whose commands require accounting. For example, to configure the Brocade device to perform TACACS+ accounting for the commands available at the Super User privilege level (that is; all commands on the device), enter the following command.

```
device(config)#aaa accounting commands 0 default start-stop tacacs+
```

An Accounting Start packet is sent to the TACACS+ accounting server when a user enters a command, and an Accounting Stop packet is sent when the service provided by the command is completed.

> NOTE
> If authorization is enabled, and the command requires authorization, then authorization is performed before accounting takes place. If authorization fails for the command, no accounting takes place.

Syntax: aaa accounting commands *privilege-level* **default start-stop** [**radius** | **tacacs+** | **none**]

The *privilege-level* parameter can be one of the following:

- **0** - Records commands available at the Super User level (all commands)
- **4** - Records commands available at the Port Configuration level (port-config and read-only commands)
- **5** - Records commands available at the Read Only level (read-only commands)

Configuring TACACS+ accounting for system events

You can configure TACACS+ accounting to record when system events occur on the Brocade device. System events include rebooting and when changes to the active configuration are made.

The following command causes an Accounting Start packet to be sent to the TACACS+ accounting server when a system event occurs, and a Accounting Stop packet to be sent when the system event is completed.

```
device(config)#aaa accounting system default start-stop tacacs+
```

Syntax: aaa accounting system default start-stop [**radius** | **tacacs+** | **none**]

Configuring an interface as the source for all TACACS and TACACS+ packets

You can designate the lowest-numbered IP address configured an Ethernet port, loopback interface, or virtual interface as the source IP address for all TACACS/TACACS+ packets from the Layer 3 Switch.

Displaying TACACS/TACACS+ statistics and configuration information

The **show aaa** command displays information about all TACACS+ and RADIUS servers identified on the device.

```
device#show aaa
Tacacs+ key: foundry
Tacacs+ retries: 1
Tacacs+ timeout: 15 seconds
Tacacs+ dead-time: 3 minutes
```

```
Tacacs+ Server: 10.95.6.90 Port:49:
                opens=6 closes=3 timeouts=3 errors=0
                packets in=4 packets out=4
no connection
Radius key: networks
Radius retries: 3
Radius timeout: 3 seconds
Radius dead-time: 3 minutes
Radius Server: 10.95.6.90 Auth Port=1812 Acct Port=1813:
                opens=2 closes=1 timeouts=1 errors=0
                packets in=1 packets out=4
no connection
```

The following table describes the TACACS/TACACS+ information displayed by the **show aaa** command.

TABLE 11 Output of the show aaa command for TACACS/TACACS+

Field	Description
Tacacs+ key	The setting configured with the **tacacs-server key** command. At the Super User privilege level, the actual text of the key is displayed. At the other privilege levels, a string of periods (....) is displayed instead of the text.
Tacacs+ retries	The setting configured with the **tacacs-server retransmit** command.
Tacacs+ timeout	The setting configured with the **tacacs-server timeout** command.
Tacacs+ dead-time	The setting configured with the **tacacs-server dead-time** command.
Tacacs+ Server	For each TACACS/TACACS+ server, the IP address, port, and the following statistics are displayed: • opens - Number of times the port was opened for communication with the server • closes - Number of times the port was closed normally • timeouts - Number of times port was closed due to a timeout • errors - Number of times an error occurred while opening the port • packets in - Number of packets received from the server • packets out - Number of packets sent to the server
connection	The current connection status. This can be "no connection" or "connection active".

The **show web connection** command displays the privilege level of Web Management Interface users.

Example

```
Brocade#show web-connection
We management Sessions:
User Privilege IP address MAC address Timeout(secs) Connection
roy READ-WRITE 10.1.1.3 0030.488.b84d9 279 HTTPS
```

Syntax: show web connection

Use the following command to clear web connections:

```
Brocade#clear web-connection
```

Syntax: clear web connection

After issuing the **clear web connection** command, the **show web connection** command displays the following output:

```
Brocade#show web-connection
No WEB-MANAGEMENT sessions are currently established!
```

HTTPS Certificates

HTTPS certificate overview

In public key cryptography each device has a pair of keys: a public key and a private key. These are typically numbers that are chosen to have a specific mathematical relationship.

The private key can be used to create a digital signature for any piece of data using a digital signature algorithm. This typically involves taking a cryptographic hash of the data and operating on it mathematically using the private key. Any device with the public key can check that this signature was created using the private key and the appropriate signature validation algorithm.

Brocade Network OS supports DSA, RSA and ECDSA encryption keys for HTTPS cryptography. You can generate key pairs, create trust points, and then authenticate and enroll the key pairs into the trust points to obtain the identity certificates.

Configuring HTTPS certificates

In order to support HTTPS, the switch needs to be configured with an Identity certificate. This task generates the key pair, then configures the trust points and certificates required for HTTPS security.

When the Apache (web server) boots. it enables HTTPS service only in the presence of HTTPS crypto certificates.

HTTP and HTTPS are mutually exclusive.

The labels for the trust point and the key pair have to be consistent throughout this process.

1. Enter Rbridge ID configuration mode.

```
switch(config)# rbridge-id 1
```

2. Generate a key pair (either RSA, ECDSA, or DSA) to sign and encrypt the security payload during the security protocol exchanges with the **crypto key** command.

```
switch(config-rbridge-id-1)# crypto key label k1 rsa modulus 2048
```

3. Configure a trusted Certificate Authority (CA) so that the imported identity certificate can be verified that it was issued by one of the locally trusted CAs with the **crypto ca** command.

```
switch(config-rbridge-id-1)# crypto ca trustpoint t1
switch(config-ca-t1)#
```

4. Associate the key pair to the trust point with the **keypair** command. 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.

```
switch(config-ca-t1)# keypair k1
```

5. Return to privileged EXEC mode with the **end** command.

```
switch(config-ca-t1)# end
```

6. You must authenticate the VDX device to the CA by obtaining the self-signed certificate of the CA with the **crypto ca authenticate** command. Because the certificate of the CA is self-signed, the public key of the CA should be manually authenticated by contacting the CA administrator to compare the fingerprint of the CA certificate.

```
switch# crypto ca authenticate t1 protocol SCP host 10.70.12.102 user fvt directory /users/home/
crypto file cacert.pem
Password: **********
```

7. Export the enrollment certificate to the location specified for the remote host with the **crypto ca enroll** command.

```
switch# crypto ca enroll t1 country US state CA locality SJ organization BRC orgunit SFI common
myhost.brocade.com protocol SCP host 10.70.12.102 user fvt directory /users/home/crypto
Password: **********
```

8. Import the identity certificate from the trust point CA with the **crypto ca import** command. This installs the identity certificate on the device.

```
switch# crypto ca import t1 certificate protocol SCP host 10.70.12.102 user fvt directory /users/
home/crypto file swcert.pem
Password: **********
```

9. Save the running configuration to the startup configuration with the **copy** command.

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

10. Confirm the configuration with the **show** commands in the example below.

```
switch# show crypto key mypubkey
rbridge-id:1
key type: rsa
key label: k1
key size: 2048

switch#  show crypto ca trustpoint
rbridge-id:1
trustpoint: t1; key-pair: k1
certificate: none
CA certificate:
SHA1 Fingerprint=76:5B:D4:2C:CB:54:FE:6B:C5:E0:E3:FD:11:B0:88:70:80:12:C6:63
Subject: C=US, ST=CA, L=SJ, O=BR, OU=SF, CN=SOUND/emailAddress=sravi
Issuer: C=US, ST=CA, L=SJ, O=BR, OU=SF, CN=SOUND/emailAddress=sravi
Not Before: Sep 19 20:56:49 2014 GMT
Not After : Oct 19 20:56:49 2014 GMT
purposes: sslserver

switch# show running-config rbridge-id crypto
rbridge-id 1
crypto key label k1 rsa modulus 2048
crypto ca trustpoint t1
keypair k1
```

11. The HTTP server (either web server or apache server) must be restarted to activate the HTTPS service. Use only one of the following methods:
    - If HTTP is in enable state (by default HTTP is enabled), then execute **http server shutdown**, followed by **no http server shutdown** to enable HTTPS.
    - If HTTP is in disable state, then execute **no http server shutdown**.
    - Reboot the switch.
    - Force an HA failover.

Disabling HTTPS certificates

Disables key pairs and trust points for HTTPS cryptography certificates, which disables the HTTPS security protocol.

To shutdown the HTTPS service without disabling the HTTPS certificates, execute the **http server shutdown** command.

When the Apache (web server) boots. it enables HTTPS service only in the presence of HTTPS crypto certificates.

HTTP and HTTPS are mutually exclusive.

> NOTE
> HTTPS certificates must be configured and enabled for web service to function on the device.

1. Delete the identity switch certificate with the **no crypto ca import** command.

```
switch# no crypto ca import t1 certificate

switch# show crypto ca certificates
rbridge-id:1
Trustpoint: t1
certificate: none
CA certificate:
SHA1 Fingerprint=76:5B:D4:2C:CB:54:FE:6B:C5:E0:E3:FD:11:B0:88:70:80:12:C6:63
Subject: C=US, ST=CA, L=SJ, O=BR, OU=SF, CN=SOUND/emailAddress=sravi
Issuer: C=US, ST=CA, L=SJ, O=BR, OU=SF, CN=SOUND/emailAddress=sravi
Not Before: Sep 19 20:56:49 2014 GMT
Not After : Oct 19 20:56:49 2014 GMT
purposes: sslserver
```

2. Unauthenticate the trust point with the **no crypto ca authenticate** command.

```
switch# no crypto ca authenticate t1

switch# show crypto ca certificates
rbridge-id:1
Trustpoint: t1
certificate: none
CA certificate: none
```

3. Disassociate the trust point from the key pair with the **no keypair** command.

```
switch(config-rbridge-id-1)# crypto ca trustpoint t1
switch(config-ca-t1)#no keypair
switch(config-ca-t1)# do show running-config rbridge-id crypto
rbridge-id 1
 crypto key label k1 rsa modulus 2048
 crypto ca trustpoint t1
  !
!
switch(config-ca-t1)# do show crypto ca trustpoint
rbridge-id:1
trustpoint: t1; key-pair: none
```

4. Delete the trust point with the **no crypto ca trustpoint** command.

```
switch(config-rbridge-id-1)# no crypto ca trustpoint t1

switch(config-ca-t1)# do show running-config rbridge-id crypto
rbridge-id 1
 crypto key label k1 rsa modulus 2048
!
switch# show crypto ca trustpoint
rbridge-id:1
trustpoint: none; key-pair: none
```

5. Delete the key pair with the **no crypto key** command.

```
switch(config-ca-t1)# exit
switch(config-rbridge-id-1)#no crypto key label k1
switch(config-rbridge-id-1)# do show running-config rbridge-id crypto
% No entries found.

switch(config-rbridge-id-1)# do show crypto key mypubkey
rbridge-id:1
key type: none
key label: none
key size: none
```

6. Return to privileged EXEC mode with the **exit** command.

```
switch(config-ca-t1)# exit
```

7. Save the running configuration to the startup configuration with the **copy** command.

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

Enabling HTTPS service

After installing the HTTPS certificates, the web server (also known as the apache server) must be restarted to configure the HTTPS service. By default, the web service is running when the device boots.

The HTTPS certificates must be installed.

The web service can be started using one of the following mechanisms:

- Restart the web service with the **http server shutdown** command, followed by the **no http server shutdown** command.
- Reboot the entire device.
- Commit an HA failover, if that option is available.

Disabling HTTPS service

The HTTPS service is disabled with the **http server shutdown** command.

ACLs

ACL overview

An access control list (ACL) is a container for rules that permit or deny network traffic based on criteria that you specify.

When a frame or packet is received or sent, the device compares its header fields against the rules in applied ACLs. This comparison is done according to a rule sequence, which you can specify. Based on the comparison, the device either forwards or drops the frame or packet.

The benefits of ACLs include the following:

- Provide security and traffic management.
- Monitor network and user traffic.
- Save network resources by classifying traffic.
- Protect against denial of service (DOS) attacks.
- Reduce debug output.

Regarding the range of filtering options, there are two types of ACL:

- *Standard ACLs* — Permit or deny traffic according to source address only.
- *Extended ACLs* — Permit or deny traffic according to source and destination addresses, as well as other parameters. For example, in an extended ACL, you can also filter by one or more of the following:
  - Port name or number
  - Protocol, for example TCP or UDP
  - TCP flags

Regarding layer and protocol, ACL types are as follows:

- Layer 2
  - MAC ACLs
- Layer 3
  - IPv4 ACLs
  - IPv6 ACLs

ACL application-targets

ACLs that you apply to interfaces, to overlay gateways, or at RBridge-level are summarized in a table. Additional ACL types, not discussed in the current unit, are described in a separate table.

The following table summarizes details of the ACL application-target types discussed in the current unit. You create all of these ACL types using the { **mac** | **ip** | **ipv6** } **access-list** command.

TABLE 12 ACLs applied to interfaces, overlay gateways, or RBridges

Target/type	Description	Applied from	Applied with	Types supported	Reference
Interface	Filters all traffic entering or exiting an interface.	Interface configuration sub-modes	{ **mac** | **ip** | **ipv6** } **access-group** { **in** | **out** }	MAC, IPv4, IPv6 Standard, extended	Layer 2 (MAC) ACLs on page 86 Layer 3 (IPv4 and IPv6) ACLs on page 91
Overlay gateway	Filters all traffic entering an overlay gateway.	Overlay-gateway configuration mode	{ **mac** | **ip** | **ipv6** } **access-group in**	MAC, IPv4, IPv6 Standard, extended	Layer 2 (MAC) ACLs on page 86 Layer 3 (IPv4 and IPv6) ACLs on page 91 *Network OS Layer 2 Switching Configuration Guide* > "VXLAN Overlay Gateways for NSX Controller Deployments"
Receive-path	Receive-path ACLs (rACLs) are applied at RBridge-level. Their primary function is to filter traffic to the route-processor CPU.	RBridge-ID configuration mode	{ **ip** | **ipv6** } **receive access-group in**	IPv4, IPv6 Standard, extended	Implementation flow for rACLs and interface ACLs on page 92

The following table summarizes details of ACL types not discussed in the current unit, as they differ significantly from ACLs applied to interfaces, overlay gateways, and RBridges.

TABLE 13 Other ACL types

Target/type	Description	Created with	Applied with	Types supported	Reference
SNMP	Restricts access to a device by IP addresses associated with an SNMP-server community or user.	{ **ip** | **ipv6** } **access-list**	(IPv4) **snmp-server community** (IPv6) **snmp-server user**	IPv4, IPv6 Standard	*Network OS Administration Guide* > "SNMP" > "Managing SNMP access rights using ACLs"
ARP	Address-resolution protocol (ARP) ACLs, applied to untrusted VLAN/VE ports to permit only ARP packets with specified IP/MAC address bindings.	**arp access-list**	**ip arp inspection filter**	There is only one type of ARP ACL.	*Network OS Security Configuration Guide* > "Configuring Dynamic ARP Inspection (DAI)" > "Implementing ARP ACLs for DAI"

NOTE

Both Layer 2 and Layer 3 ACLs are supported under flow-based QoS. For more information, refer to the "QoS" > "Flow-based QoS" section of the *Network OS Layer 2 Switching Configuration Guide*.

Interface ACLs and rACLs

Layer 3 ACLs applied at RBridge-level to filter route-processor CPU traffic are called *receive-path ACLs* or *rACLs*. All other ACLs discussed in the current document are applied to an interface or to an overlay gateway. They can be referred to an *interface ACLs*.

Traffic entering an RBridge can be divided into two categories:

- Datapath traffic
- Traffic for the route-processor CPU

Rules in an ACL applied to an interface filter all traffic entering or exiting that interface—datapath traffic and route-processor traffic.

Rules in an rACL, applied at RBridge level, primarily filter traffic destined for the route-processor CPU. Implementing rACLs offers the following advantages:

- Shields the route-processor CPU from unnecessary and potentially harmful traffic.
- Mitigates denial of service (DoS) attacks.
- Protects the CPU by a single application, rather than needing to apply ACLs on multiple interfaces.

rACLs also support filtering multicast datapath traffic, which offers an alternative to applying ACLs containing multicast rules to all device interfaces.

To implement rACLs, refer to Implementation flow for rACLs and interface ACLs on page 92.

Otherwise, continue with ACLs applied to interfaces on page 85.

ACLs applied to interfaces

This topic describes interfaces and overlay gateways that support ACLs.

Layer 2 (MAC) ACLs are supported on the following user-interface types:

- Physical interfaces (<*N*>-gigabit Ethernet)—in switchport mode
- Port-channel interfaces—in switchport mode
- VLANs
- Overlay gateways

Layer 3 (IPv4 and IPv6) ACLs are supported on the following interface types:

- User interfaces
  - Physical interfaces (<*N*>-gigabit Ethernet)
  - Port-channel interfaces
  - Virtual Ethernet (VE) interfaces
- Management interfaces
- Overlay gateways

ACL and rule limits

There are limits to the number of ACLs and rules supported.

The following table lists ACL and rule limits for supported devices and ACL types:

TABLE 14 ACL and rule limits

Resource	Brocade VDX 6740 Brocade VDX 6940	Brocade VDX 8770
Maximum total MAC ACLs (standard and extended)	512	2048
Maximum rules per MAC ACL	Total rules: 256 Maximum **count** rules: 256	Total rules: 2048 Maximum **count** rules: 2048
Maximum total IPv4 ACLs (standard and extended)	512	2048

TABLE 14 ACL and rule limits (continued)

Resource	Brocade VDX 6740 Brocade VDX 6940	Brocade VDX 8770
Maximum rules per IPv4 ACL	Total rules: 256 Maximum **count** rules: 256	Total rules: 12288 Maximum **count** rules: 6144
Maximum total IPv6 ACLs (standard and extended)	512	2048
Maximum rules per IPv6 ACL	Total rules: 256 Maximum **count** rules: 256	Total rules: 2048 Maximum **count** rules: 2048
Maximum total rules supported (All ACL rules on device)	200K	200K

The following limits apply to every ACL:

- An ACL name can be 1 through 63 characters long, and must begin with a–z, A–Z or 0–9. You can also use underscore (_) or hyphen (-) in an ACL name, but not as the first character.
- Sequence numbers can range from 0 through 4294967290.

Layer 2 (MAC) ACLs

MAC ACL configuration guidelines

Follow these guidelines and restrictions when configuring MAC ACLs.

- On any given device, an ACL name must be unique among all ACL types (MAC/IPv4/IPv6; standard or extended).
- The order of the rules in an ACL is critical. The first rule that matches the traffic stops further processing of the frames. For example, following an **apply** match, subsequent **deny** or **hard-drop** rules do not override the **apply**.
- When you add rules to an ACL, you have the option of specifying the rule sequence number. If you create a rule without a sequence number, it is automatically assigned a sequence number incremented above the previous last rule.
- There is an implicit "permit" rule at the end of every Layer 2 rules list. This permits all Layer 2 streams that do not match any of the "deny" rules in the ACL.
- If an ACL includes a rule that denies a specific host or range, the device still responds to the **ping** command, unless there is also a relevant rule with the **hard drop** option.
- A hard-drop rule overrides control packet trap entries. As a result, it may interfere with normal operations of the protocols.
- Existing ACL rules cannot be changed, or updated with elements like **count** and **log**. You need to delete the rule and recreate it with the changed elements.
- You can apply up to six ACLs to each user interface, as follows:
  - One ingress MAC ACL—if the interface is in switchport mode
  - One egress MAC ACL—if the interface is in switchport mode
  - One ingress IPv4 ACL
  - One egress IPv4 ACL
  - One ingress IPv6 ACL
  - One egress IPv6 ACL

NOTE
You can apply a specific ACL to one or more interfaces, for ingress or egress, or for both.

Guidelines for ACLs applied to overlay gateways

In addition to the general guidelines, the following additional guidelines are relevant for ACLs applied to overlay gateways:

- There is an implicit "deny" rule at the end of every ACL applied to an overlay gateway. This denies all streams that do not match any of the "permit" rules in the ACL.
- You can apply a maximum of three ACLs to an overlay gateway, as follows:
  - One ingress MAC ACL
  - One ingress IPv4 ACL
  - One ingress IPv6 ACL

Creating a standard MAC ACL

A standard ACL permits or denies traffic according to source address only.

1. Enter **configure** to access global configuration mode.

```
device# configure
```

2. Enter the **mac access-list standard** command to create the ACL.

```
device(config)# mac access-list standard test_01
device(conf-macl-std)#
```

3. For each ACL rule that you need to create, enter a permit or deny command, specifying the needed parameters.

```
device(conf-macl-std)# seq 100 deny host 0011.2222.3333 count
device(conf-macl-std)# seq 110 permit host 0022.1111.2222 ffff.ffff.00ff count
device(conf-macl-std)# deny host 0022.3333.4444 count
device(conf-macl-std)# permit host 0022.5555.3333 count
```

4. Apply the ACL that you created to the appropriate interface.

Creating an extended MAC ACL

An extended ACL permits or denies traffic according to one or more of the following parameters: source address, destination address, port, ethertype, VLAN.

1. Enter **configure** to access global configuration mode.

```
device# configure
```

2. Enter the **mac access-list extended** command to create the access list.

```
device(config)# mac access-list extended test_02
```

3. Create a rule in the MAC ACL to **permit** traffic with the source MAC address and the destination MAC address.

```
device(conf-macl-ext)# permit host 0022.3333.4444 host 0022.3333.5555
```

4. (Optional) Use the **seq** command to insert the rule anywhere in the MAC ACL.

```
device(conf-macl-ext)# seq 5 permit host 0022.3333.4444 host 0022.3333.5555
```

5. Apply the ACL that you created to the appropriate interface.

Applying Layer 2 ACLs to interfaces

An ACL affects network traffic only after you apply it to an interface, using an **access-group** command. Use these procedures to apply MAC standard or extended ACLs to interfaces.

Applying a MAC ACL to a physical interface

Use this procedure to apply a Layer 2 ACL to any physical interface.

1. Enter the **configure** command to access global configuration mode.

```
device# configure
```

2. Enter the **interface** command, specifying the interface type and the rbridge-id/slot/port number.

```
device(config)# interface tengigabitethernet 2/2/1
```

3. If needed, to configure the interface as a Layer 2 switch port, enter the **switchport** command.
4. Enter the **mac access-group** command, specifying the ACL that you are applying to the interface, the in/out direction, and (optionally) **routed** or **switched**.

```
device(conf-if-te-2/2/1)# mac access-group test_02 in
```

Applying a MAC ACL to a LAG interface

Use this procedure to apply a Layer 2 ACL to a LAG (logical) interface, in switchport mode.

1. Enter the **configure** command to access global configuration mode.

```
device# configure
```

2. Enter the **interface port-channel** command, specifying the port-channel number.

```
device(config)# interface port-channel 10
```

3. Enter the **mac access-group** command, specifying the ACL that you are applying to the interface, the in/out direction, and (optionally) **routed** or **switched**.

```
device(config-Port-channel-10)# mac access-group test_02 in
```

Applying a MAC ACL to a VLAN interface

Use this procedure to apply a Layer 2 ACL to a VLAN interface.

1. Enter the **configure** command to access global configuration mode.

```
device# configure
```

2. Enter the **interface vlan** command, specifying the *vlan-id*.

```
device(config)# interface vlan 50
```

3. Enter the **mac access-group** command, specifying the ACL that you are applying to the interface, the in/out direction, and (optionally) **routed** or **switched**.

```
device(config-Vlan-50)# mac access-group test_02 in
```

Applying a MAC ACL to an overlay gateway

Use this procedure for applying a Layer 2 ACL to a VXLAN overlay gateway.

1. Enter the **configure** command to access global configuration mode.

```
device# configure
```

2. Enter the **overlay-gateway** command to access VXLAN overlay-gateway configuration mode for a gateway that you configure.

```
device(config)# overlay-gateway gw121
```

3. Enter the **mac access-group** command, specifying the ACL and **in**.

```
device(config-overlay-gw-gw121)# mac access-group stdmacaclin in
```

Removing a MAC ACL

To suspend ACL rules, you can remove the ACL containing those rules from the interface to which it was applied. After removing it, you can also delete the ACL.

1. Enter the **configure** command to access global configuration mode.

```
device# configure
```

2. Enter the **interface** command, specifying the interface type and identifying number.

```
device(config)# interface tengigabitethernet 178/0/9
```

3. Enter the **no access-group** command.

```
device(conf-if-te-178/0/9)# no mac access-group macacl2 in
```

Modifying MAC ACL rules

To modify an ACL rule, delete the original rule and replace it with a new rule.

1. To display MAC ACL rule details, in privileged EXEC mode enter the **show running-config mac access-list** command.

```
device# show running-config mac access-list standard ACL1
mac access-list standard ACL1
  seq 100 deny host 0022.3333.4444 count
  seq 110 permit host 0011.3333.5555 count
```

Note the **seq** number of the rule that you need to modify.

2. Enter the **configure** command to access global configuration mode.

```
device# configure
```

3. Enter the **mac access-list** command, specifying the ACL you need to modify.

```
device(config)# mac access-list standard ACL1
```

4. Delete the original rule, doing one of the following:
   - Enter the **no seq** command, specifying the sequence number of the rule that you are deleting.

```
device(conf-macl-std)# no seq 100
```

   - Enter the exact rule that you are deleting, preceded by **no**.

```
no deny host 0022.3333.4444 count
```

5. Enter the replacement rule.

```
device(conf-macl-ext)# seq 100 permit host 0022.3333.6666 count
```

Reordering the sequence numbers in a MAC ACL

Reordering ACL-rule sequence numbers is helpful if you need to insert new rules into an ACL in which there are not enough available sequence numbers.

Note the following regarding sequence numbers and their reordering parameters:

- The default initial sequence number is 10 and the default increment is 10.
- For reordering the sequence numbers, you need to specify the following:
  - The new starting sequence number
  - The increment between sequence numbers

The first rule receives the number of the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment number that you specify. The starting-sequence number can range from 0 through 4294967290, and the increment number can range from 1 through 4294967290.

For example: In the command below, the **resequence access-list** command assigns a sequence number of 50 to the first rule, 55 to the second rule, 60 to the third rule, and so forth.

```
device# resequence access-list mac test_02 50 5
```

Creating a MAC ACL rule enabled for counter statistics

When you create ACL rules, the **count** parameter enables you to display counter statistics.

1. Enter the **configure** command to access global configuration mode.

```
device# configure
```

2. Enter the **mac access-list** command to create or modify an access list.

```
device(config)# mac access-list standard mac_acl_1
```

3. In each rule for which you need to display statistics, include the **count** keyword.

```
device(conf-mac1-std)# seq 100 deny 0022.3333.4444 count
```

4. If you have not yet applied the ACL to the appropriate interface, do so now.
5. (Optional) To display ACL counter statistics, enter the **show statistics access-list** command.

ACL logs

ACL logs can provide insight into permitted and denied network traffic.

ACL logs maintain the following properties:

- Supported for all ACL types (MAC, IPv4, and IPv6)
- Supported for incoming and outgoing network traffic
- Supported for all user interfaces (but not on management interfaces) on which ACLs can be applied
- May be CPU-intensive

Enabling and configuring the ACL log buffer

Among the conditions required for ACL logging is that the ACL log buffer be enabled and configured.

1. Enter the **debug access-list-log buffer** command to enable and configure ACL log buffering.

```
device# debug access-list-log buffer circular packet count 1600
```

2. (Optional) To display the current ACL log buffer configuration, enter the **show access-list-log buffer config** command.

```
device# show access-list-log buffer config
ACL Logging Enabled.
ACL logging Buffer configuration: Buffer type is circular and Buffer size is 1600.
```

Creating a MAC ACL rule enabled for logging

When you create ACL rules for which you want to enable logging, you must include the **log** keyword.

1. Enter the **configure** command to access global configuration mode.

```
device# configure
```

2. Enter the **mac access-list** command to create or modify an access list.

```
device(config)# mac access-list standard mac_1
```

3. In each rule for which you need logging, include the **log** keyword.

```
device(conf-mac1-std)# seq 100 deny 0022.3333.4444 log
```

4. If you have not yet applied the ACL to the appropriate interface, do so now.
5. (Optional) To display ACL logs, enter the **show access-list log buffer** command.

Layer 3 (IPv4 and IPv6) ACLs

Layer 3 access control lists (ACLs) filter traffic based on IPv4 or IPv6 header fields.

Implementation flow for rACLs and interface ACLs

The implementation flows for Layer 3 interface ACLs (including ACLs applied to overlay gateways) and receive-path ACLs (rACLs) are similar.

NOTE
For a comparison of rACLs and interface ACLs, refer to Interface ACLs and rACLs on page 84.

The following table displays the differential flows of implementation topics for interface ACLs and rACLs:

Interface ACLs	All ACLs	rACLs
	Layer 3 ACL configuration guidelines on page 92	
	One of the following procedures: • Creating a standard IPv4 ACL on page 95 • Creating a standard IPv6 ACL on page 95 • Creating an extended IPv4 ACL on page 95 • Creating an extended IPv6 ACL on page 96	
Applying Layer 3 ACLs to interfaces on page 97		Applying Layer 3 rACLs to RBridges on page 99

The above table indicates that there are no structural differences between Layer 3 interface ACLs and rACLs; you use identical procedures for all types. The implementation differences are as follows:

- You apply interface ACLs from an interface configuration mode, and overlay-gateway ACLs from overlay-gateway mode, all using the { **ip** | **ipv6** } **access-group** { **in** | **out** } command.
- You apply rACLs from rbridge-id configuration mode, using the { **ip** | **ipv6** } **receive access-group in** command.

All of the following topics apply both to interface ACLs and to rACLs:

- Modifying Layer 3 ACL rules on page 100
- Reordering the sequence numbers in a Layer 3 ACL on page 100
- ACL counter statistics (Layer 3) on page 101
- ACL logs on page 91
- ACL Show and Clear commands on page 103

Layer 3 ACL configuration guidelines

We present guidelines for all Layer 3 ACLs, followed by guidelines for ACLs applied to a user interface, applied to a management interface, applied to an overlay gateway, and then guidelines for receive-path ACLs (rACLs).

The following are guidelines for all Layer 3 ACLs:

- An ACL name can be up to 63 characters long, and must begin with a–z, A–Z or 0–9. You can also use underscore (_) or hyphen (-) in an ACL name, but not as the first character.
- On any given switch, an ACL name must be unique among all ACL types (MAC/IPv4/IPv6, standard or extended, general or receive).

- The order of the rules in an ACL is critical. The first rule that matches the traffic stops further processing of the frames. For example, following an **apply** match, subsequent **deny** or **hard-drop** rules do not override the **apply**.
- When you create an ACL rule, you have the option of specifying the rule sequence number. If you create a rule without a sequence number, it is automatically assigned a sequence number incremented above the previous last rule.
- Existing ACL rules cannot be changed, or updated with elements like **count** and **log**. You need to delete the rule and recreate it with the changed elements.
- If an ACL includes a rule that denies a specific host or range (for example: "seq 2 deny host 10.9.106.120"), the switch still responds to the **ping** command, unless there is also a relevant rule with the **hard drop** option (such as `seq 20 hard-drop icmp any any`).
- A hard-drop rule overrides control packet trap entries. As a result, it may interfere with normal operations of the protocols.
- If—under IPv6—RA-Guard is enabled on an interface, there is an internal rule that takes precedence over user-configured rules applied to that interface. For example:

```
seq 10 hard-drop IPv6-ICMP any any icmp-type 134 icmp-code 0
```

Guidelines for Layer 3 ACLs applied to user interfaces

In addition to the general guidelines, the following additional guidelines are relevant for Layer 3 ACLs applied to user interfaces:

- There is an implicit "deny" rule at the end of every Layer 3 ACL applied to a user interface. This denies all L3 streams that do not match any of the "permit" rules in the ACL.
- You can apply a maximum of six ACLs to a user interface, as follows:
  - One ingress MAC ACL—if the interface is in switchport mode
  - One egress MAC ACL—if the interface is in switchport mode
  - One ingress IPv4 ACL
  - One egress IPv4 ACL
  - One ingress IPv6 ACL
  - One egress IPv6 ACL

> NOTE
> You can apply a specific ACL to one or more interfaces, for ingress or egress, or for both.

Guidelines for ACLs applied to a management interface

In addition to the general guidelines, the following additional guidelines are relevant for Layer 3 ACLs applied to a management interface:

- For an ACL applied to a management interface, Layer 3 streams that do not match any of the "deny" rules in the ACL are permitted.
- For an ACL applied to a management interface, **hard-drop** parameters are interpreted as **deny** parameters.
- (Extended ACLs) Applying a permit or deny UDP ACL to the management interface enacts an implicit deny for TCP; however, a ping will succeed.
- (Extended ACLs) Applying a permit or deny ACL for a specific UDP port enacts an implicit deny for all other UDP ports.
- (Extended ACLs) Applying a permit or deny ACL for a specific TCP port enacts an implicit deny for all other TCP ports.
- ACLs in a route-map are not used by the Open Shortest Path First (OSPF) or Border Gateway Protocol (BGP) protocols.

- You can apply a maximum of two ACLs to a management interface, as follows:
  - One ingress IPv4 ACL
  - One ingress IPv6 ACL
- Before downgrading firmware, unbind any ACLs on the management interface.

If no ACLs are applied to the switch management interface, the following default rules are effective:

- seq 0 permit tcp any any eq 22
- seq 1 permit tcp any any eq 23
- seq 2 permit tcp any any eq 80
- seq 3 permit tcp any any eq 443
- seq 4 permit udp any any eq 161
- seq 5 permit udp any any eq 123
- seq 6 permit tcp any any range 600-65535
- seq 7 permit udp any any range 600-65535

Guidelines for ACLs applied to overlay gateways

In addition to the general guidelines, the following additional guidelines are relevant for ACLs applied to overlay gateways:

- There is an implicit "deny" rule at the end of every ACL applied to an overlay gateway. This denies all streams that do not match any of the "permit" rules in the ACL.
- You can apply a maximum of three ACLs to an overlay gateway, as follows:
  - One ingress MAC ACL
  - One ingress IPv4 ACL
  - One ingress IPv6 ACL

Guidelines for receive-path ACLs (rACLs)

In addition to the general guidelines, the following additional guidelines are relevant for rACLs:

- Interface ACLs and rACLs share the same resource (database-table).
- To drop CPU-bound traffic, specify the **hard-drop** option. **Permit** and **deny** both allow CPU-bound traffic.
- IPv4 rACLs apply to multicast datapath traffic only if multicast destination-IPs are explicitly specified in rules.
- In an IPv4 rACL rule, if a destination IP or **any** is not specified, *my-ip* (IP addresses configured on any Layer 3 interface) is interpreted as the destination IP. Such rules do not filter multicast traffic.
- By default, IPv6 rACLs apply both to route-processor CPU traffic and to multicast datapath traffic. Unicast datapath traffic is not affected by rACLs.
- If in an IPv6 rACL rule a destination IP is not specified, the destination IP is interpreted both as *my-ip* and as multicast IP.
- Multicast traffic is first filtered by rACLs, then by interface ACLs.
- In all rACLs, explicit and implicit rules are processed in the following order:
  1. Explicit rules, in an order determined by their **seq** numbers.
  2. An implicit **permit** rule for all Layer 3 control protocols.
  3. An implicit **hard-drop any** *my-ip* rule that affects all other CPU-bound traffic.
- Under inband management, you need to include permit rules for your telnet/SSH access to the device.

Creating a standard IPv4 ACL

A standard ACL permits or denies traffic according to source address only.

1. Enter **configure** to access global configuration mode.

```
switch# configure
```

2. Enter the **ip access-list standard** command to create the access list.

```
switch(config)# ip access-list standard stdACL3
```

3. For each ACL rule, enter a **seq** command, specifying the needed parameters.

```
switch(config-ipacl-std)# seq 5 permit host 10.20.33.4
switch(config-ipacl-std)# seq 15 deny any
```

4. Apply the ACL that you created to the appropriate interface.

The following example shows how to create a standard IPv4 ACL, define a rule for it, and apply the ACL to an interface.

```
switch# configure
switch(config)# ip access-list standard stdACL3
switch(config-ipacl-std)# seq 5 permit host 10.20.33.4
switch(config-ipacl-std)# seq 15 deny any
switch(config-ipacl-std)# exit
switch(config)# interface tengigabitethernet 122/5/22
switch(conf-if-te-122/5/22)# ip access-group stdACL3 in
```

Creating a standard IPv6 ACL

A standard ACL permits or denies traffic according to source address only.

1. Enter **configure** to access global configuration mode.

```
switch# configure
```

2. Enter the **ipv6 access-list standard** command to create the access list.

```
switch(config)# ipv6 access-list standard std_V6_ACL4
```

3. For each ACL rule, enter a **[seq] {permit | deny | hard-drop}** command, specifying the needed parameters.

```
switch(config-ip6acl-std)# seq 5 permit host 2001:db8::1:2
switch(config-ip6acl-std)# seq 15 deny any
```

4. Apply the ACL that you created to the appropriate interface.

Creating an extended IPv4 ACL

An extended ACL permits or denies traffic according to one or more of the following parameters: source address, destination address, port, protocol (TCP or UDP), TCP flags.

1. Enter **configure** to access global configuration mode.

```
switch# configure
```

2. Enter the **ip access-list extended** command to create the access list.

```
switch(config)# ip access-list extended extdACL5
```

3. For each ACL rule, enter a **[seq] {permit | deny | hard-drop}** command—specifying the needed parameters.

```
switch(config-ipacl-ext)# seq 5 deny tcp host 10.24.26.145 any eq 23
switch(config-ipacl-ext)# seq 7 deny tcp any any eq 80
switch(config-ipacl-ext)# seq 10 deny udp any any range 10 25
switch(config-ipacl-ext)# seq 15 permit tcp any any
```

4. Apply the ACL that you created to the appropriate interface.

The following example creates an IPv4 extended ACL, defines rules in the ACL, and applies it as a receive-path ACL to an RBridge.

```
device(config)# ip access-list extended ipv4-receive-acl-example
device(conf-ipacl-ext)# hard-drop tcp host 10.0.0.1 any count
device(conf-ipacl-ext)# hard-drop udp any host 20.0.0.1 count
device(conf-ipacl-ext)# permit tcp host 10.0.0.2 any eq telnet count
device(conf-ipacl-ext)# permit tcp host 10.0.0.2 any eq bgp count
device(conf-ipacl-ext)# hard-drop tcp host 10.0.0.3 host 224.0.0.1 count

device(conf-ipacl-ext)# rb 1
device(config-rbridge-id-1)# ip receive access-group ipv4-receive-acl-example in
```

Creating an extended IPv6 ACL

An extended ACL permits or denies traffic according to one or more of the following parameters: source address, destination address, port, protocol (TCP or UDP), TCP flags.

1. Enter **configure** to access global configuration mode.

```
switch# configure
```

2. Enter the **ipv6 access-list extended** command to create the access list.

```
switch(config)# ipv6 access-list extended ip_acl_1
```

3. For each ACL rule, enter a **[seq] {permit | deny | hard-drop}** command—specifying the needed parameters.

```
switch(conf-ip6acl-ext)# seq 10 deny ipv6 2001:2002:1234:1::/64 2001:1001:1234:1::/64 count
```

4. Apply the ACL that you created to the appropriate interface.

The following example shows how to create an extended IPv6 ACL, define rules for it (including a rule that filters by DSCP ID), and apply the ACL to an interface.

```
switch# configure
switch(config)# ipv6 access-list extended ip_acl_1
switch(conf-ip6acl-ext)# seq 10 deny ipv6 any any dscp 3
switch(conf-ip6acl-ext)# seq 20 deny ipv6 2001:2002:1234:1::/64 2001:1001:1234:1::/64 count
switch(conf-ip6acl-ext)# exit
switch(config)# interface ten 122/5/22
switch(conf-if-te-122/5/22)# ipv6 access-group ip_acl_1 in
```

The following example creates an IPv6 extended ACL, defines rules in the ACL, and applies it as a receive-path ACL to an RBridge.

```
device(config)# ipv6 access-list extended ipv6-receive-acl-example
device(conf-ipacl-ext)# hard-drop tcp host 10::1 any count
device(conf-ipacl-ext)# hard-drop udp any host 20::1 count
device(conf-ipacl-ext)# permit tcp host 10::2 any eq telnet count
device(conf-ipacl-ext)# permit tcp host 10::2 any eq bgp count
device(conf-ipacl-ext)# hard-drop tcp host 10::3 host ff02::1 count

device(conf-ipacl-ext)# rb 1
device(config-rbridge-id-1)# ipv6 receive access-group ipv6-receive-acl-example in
```

Applying Layer 3 ACLs to interfaces

An ACL affects network traffic only after you apply it to an interface, using one of the **access-group** commands. Use these procedures to apply standard or extended IPv4 and IPv6 ACLs to interfaces or to remove them from the interfaces.

Applying a Layer 3 ACL to a physical interface

Use this procedure for applying an IPv4 or IPv6 ACL to a physical interface, using the **ip/ipv6 access-group** command.

1. Enter **configure** to change to global configuration mode.

```
device# configure
```

2. Enter the **interface ethernet** command, specifying the slot/port number.

```
device(config)# interface ethernet 5/2
```

3. Enter the **ip/ipv6 access-group** command, specifying the ACL that you are applying to the interface and the in/out direction.

```
device(conf-if-eth-5/2)# ipv6 access-group ip_acl_1 in
```

The following example shows how to apply an IPv6 ACL to a physical interface.

```
device# configure
device(config)# interface ethernet 5/2
device(conf-if-eth-5/2)# ipv6 access-group ip_acl_1 in

device(conf-if-eth-5/2)# do show access-list ipv6 ip_acl_1 in
ipv6 access-list ip_acl_1 on TenGigabitEthernet 122/5/22 at Ingress (From User)
    seq 10 deny ipv6 2001:2002:1234:1::/64 2001:1001:1234:1::/64 count (Active)
```

Applying a Layer 3 ACL to a LAG interface

Use this procedure to apply an IPv4 or IPv6 ACL to a LAG (logical) interface.

1. Enter the **configure** command to access global configuration mode.

```
device# configure
```

2. Enter the **interface port-channel** command, specifying the port-channel number.

```
device(config)# interface port-channel 10
```

3. Enter the **ip/ipv6 access-group** command, specifying the ACL that you are applying to the interface, the in/out direction, and (optionally) **routed** or **switched**.

```
device(config-Port-channel-10)#ip access-group test_02 in
```

Applying a Layer 3 ACL to a VE interface

Use this procedure to apply an IPv4 or IPv6 ACL to a VE interface.

1. Enter the **configure** command to access global configuration mode.

```
device# configure
```

2. Enter the **interface ve** command, specifying the *vlan-id*.

```
device(config)# interface ve 50
```

3. Enter the **ip/ipv6 access-group** command, specifying the ACL that you are applying to the VE, the in/out direction, and (optionally) **routed** or **switched**.

```
device(config-ve-50)# ip access-group test_02 in
```

Applying a Layer 3 ACL to an overlay gateway

Use this procedure for applying a Layer 3 ACL to a VXLAN overlay gateway.

1. Enter the **configure** command to access global configuration mode.

```
device# configure
```

2. Enter the **overlay-gateway** command to access VXLAN overlay-gateway configuration mode for a gateway that you configure.

```
device(config)# overlay-gateway gw121
```

3. Enter one or both of the following commands, specifying network protocol (**ip** or **ipv6**), the ACL, and **in**.

```
sw0(config-overlay-gw-gw121)# ip access-group stdipaclin in
sw0(config-overlay-gw-gw121)# ipv6 access-group stdipv6aclin in
```

Applying a Layer 3 ACL to a management interface

Use this procedure for applying a Layer 3 ACL to a management interface, using the **access-group** command.

NOTE
In virtual cluster switching (VCS) mode, you can apply an ACL to any cluster node, specifying its RBridge ID and port.

NOTE
If an explicit "deny ip any any" IP rule is applied to the management interface, that IP rule has priority over any TCP or UDP rules. Any incoming TCP packets that match that IP rule are dropped because the TCP packet has an IP header.

1. Enter **configure** to access global configuration mode.

```
switch# configure
```

2. Use the **interface management** command to enter configuration mode for the management interface, specifying RBridge ID/port.

```
switch(config)# interface management 3/1
```

3. To apply an IPv4 ACL to the management interface, enter the **ip access-group** command, specifying the ACL that you are applying to the interface, and **in**

```
switch(config-Management-3/1)# ip access-group stdACL3 in
```

4. To apply an IPv6 ACL to the management interface, enter the **ipv6 access-group** command, specifying the ACL that you are applying to the interface, and **in**.

```
switch(config-Management-3/1)# ipv6 access-group stdV6ACL1 in
```

5. Use the **exit** command to return to global configuration mode. Your changes are automatically saved.

```
switch(config-Management-3/1)# exit
```

Removing a Layer 3 ACL from an interface

To suspend ACL rules, you can remove the ACL containing those rules from the interface to which it was applied. After removal, you can also delete the ACL.

1. Enter the **configure** command to access global configuration mode.

```
switch# configure
```

2. Enter the **interface** command, specifying the interface type and name.

```
switch(config)# interface tengigabitethernet 122/5/22
```

3. Enter the **no access-group** command.

```
switch(conf-if-te-122/5/22)# no ipv6 access-group ip_acl_1 in
```

Applying Layer 3 rACLs to RBridges

A receive-path ACL (rACL) affects traffic only after you apply it at RBridge-level. Use these procedures to apply standard or extended IPv4 and IPv6 ACLs to RBridges or to remove them from the RBridges.

Applying an rACL to an RBridge

Use this procedure for applying an IPv4 or IPv6 receive-path ACL (rACL) at RBridge-level, using one of the **receive access-group** commands.

1. Enter **configure terminal** to change to global configuration mode.

```
device# configure terminal
```

2. Enter the **rbridge-id** command, specifying the RBridge ID.

```
device(config)# rbridge-id 1
```

3. Enter the { **ip** | **ipv6** } **receive access-group** command, specifying the ACL that you are applying to the RBridge and the **in** direction.

```
device(config-rbridge-id-1)# ip receive access-group ipv4-receive-acl-example in
```

The following example shows how to create an IPv6 ACL, define rules needed for an rACL, and apply the ACL to an RBridge.

```
device# configure terminal
device(config)# ipv6 access-list extended ipv6-receive-acl-example
device(conf-ipacl-ext)# hard-drop tcp host 10::1 any count
device(conf-ipacl-ext)# hard-drop udp any host 20::1 count
device(conf-ipacl-ext)# permit tcp host 10::2 any eq telnet count
device(conf-ipacl-ext)# permit tcp host 10::2 any eq bgp count
device(conf-ipacl-ext)# rbridge-id 1
device(config-rbridge-id-1)# ipv6 receive access-group ipv6-receive-acl-example in
```

Removing an rACL from an RBridge

To suspend rACL rules, you can remove the ACL containing those rules from the RBridge to which it was applied. After removal, you can also delete the ACL.

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
```

2. Enter the **rbridge-id** command, specifying the RBridge ID.

```
device(config)# rbridge-id 1
```

3. Enter the **no** { **ip** | **ipv6** } **receive access-group** command, specifying the ACL name and the **in** direction.

```
device(config-rbridge-id-1)# no ip receive access-group ipv4-receive-acl-example in
```

Modifying Layer 3 ACL rules

To modify an ACL rule, delete the original rule and replace it with a new rule.

1. To display the rules of all ACLs of a given IP type and standard/extended specification, in global configuration mode enter the **show running-config** command.

```
device# show running-config ip access-list standard
ip access-list standard a1
 seq 10 permit host 10.1.1.1 count
```

   Note the **seq** number of the rule that you need to delete or modify.

2. Enter the **configure** command to access global configuration mode.

```
device# configure
```

3. Enter the **{ip | ipv6} access-list** command, specifying the ACL you need to modify.

```
device(config)# ip access-list standard a1
```

4. Delete the original rule, doing one of the following:
   - Enter the **no seq** command, specifying the sequence number of the rule that you are deleting.

```
device(conf-ipacl-std)# no seq 10
```

   - Enter the exact rule that you are deleting, preceded by **no**.

```
no permit host 10.1.1.1 count
```

5. Enter the replacement rule.

```
device(conf-ipacl-std)# seq 10 permit host 10.1.1.1 log
```

Reordering the sequence numbers in a Layer 3 ACL

Reordering ACL-rule sequence numbers is helpful if you need to insert new rules into an ACL in which there are not enough available sequence numbers.

NOTE

Although you can use this procedure for IPv4 or IPv6 ACLs, the example is for IPv4.

Note the following regarding sequence numbers and their reordering parameters:

- The default initial sequence number is 10 and the default increment is 10.
- For reordering the sequence numbers, you need to specify the following:
  - The new starting sequence number
  - The increment between sequence numbers

The first rule receives the number of the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment number that you specify. The starting-sequence number can range from 0 through 4294967290, and the increment number can range from 1 through 4294967290.

For example: In the command below, for the IPv4 ACL "a1", the **resequence access-list** command assigns a sequence number of 5 to the first rule, 10 to the second rule, 15 to the third rule, and so forth.

```
switch# resequence access-list ip a1 5 5
```

ACL counter statistics (Layer 3)

If an ACL rule contains the **count** parameter, you can access statistics for the rule, including the number of frames permitted or denied by that rule. If needed, you can also clear ACL statistics.

NOTE
If an ACL with rules that contain the **count** keyword is applied to a management interface, statistics are not recorded for that ACL.

Creating an IPv4 ACL rule enabled for counter statistics

When you create ACL rules, the **count** parameter enables you to display counter statistics.

1. Enter **configure** to access global configuration mode.

```
device# configure
```

2. Enter the **ip access-list** command to create or modify an access list.

```
device(config)# ip access-list standard stdACL3
```

3. For each ACL rule for which you need to display statistics, include the **count** keyword.

```
device(config-ipacl-std)# seq 5 permit host 10.20.33.4 count
device(config-ipacl-std)# seq 15 deny any count
```

4. If you have not yet applied the ACL to the appropriate interface, do so now.

Creating an IPv6 ACL rule enabled for counter statistics

When you create ACL rules, the **count** parameter enables you to display counter statistics.

1. Enter the **configure** command to access global configuration mode.

```
device# configure
```

2. Enter the **ipv6 access-list** command to create or modify an access list.

```
device(config)# ipv6 access-list extended ip_acl_1
```

3. For each ACL rule for which you need to display statistics, include the **count** keyword.

```
device(conf-ip6acl-ext)# seq 20 deny ipv6 2002:2003:1234:1::/64 2001:3001:1234:1::/64 count
```

4. If you have not yet applied the ACL to the appropriate interface, do so now.
5. (Optional) To display ACL counter statistics, enter the **show statistics access-list** command.

```
device# show statistics access-list ipv6 ip_acl_1 in
ipv6 access-list ip_acl_1 on Ethernet 2/3 at Ingress (From User)
    seq 10 deny ipv6 2001:2002:1234:1::/64 2001:1001:1234:1::/64 count (0 frames)
    seq 20 deny ipv6 2002:2003:1234:1::/64 2001:3001:1234:1::/64 count (33 frames)
```

The following example shows how to create an IPv6 extended ACL and define a counter-enabled rule for it.

```
device# configure
device(config)# ipv6 access-list extended ip_acl_1
device(conf-ip6acl-ext)# seq 10 deny ipv6 2001:2002:1234:1::/64 2001:1001:1234:1::/64 count
```

ACL logs

ACL logs can provide insight into permitted and denied network traffic.

ACL logs maintain the following properties:

- Supported for all ACL types (MAC, IPv4, and IPv6)
- Supported for incoming and outgoing network traffic
- Supported for all user interfaces (but not on management interfaces) on which ACLs can be applied
- May be CPU-intensive

Enabling and configuring the ACL log buffer

Among the conditions required for ACL logging is that the ACL log buffer be enabled and configured.

1. Enter the **debug access-list-log buffer** command to enable and configure ACL log buffering.

```
device# debug access-list-log buffer circular packet count 1600
```

2. (Optional) To display the current ACL log buffer configuration, enter the **show access-list-log buffer config** command.

```
device# show access-list-log buffer config
ACL Logging Enabled.
ACL logging Buffer configuration: Buffer type is circular and Buffer size is 1600.
```

Enabling IPv6 ACL rules for logging

When you create ACL rules for which you want to enable logging, you must include the **log** parameter.

1. Enter the **configure** command to access global configuration mode.

```
device# configure
```

2. Enter the **ipv6 access-list** command to create or modify an access list.

```
device(config)# ipv6 access-list extended ipv6_acl_1
```

3. For each ACL rule for which you need logging, include the **log** keyword.

```
device(conf-ip6acl-ext)# seq 20 deny ipv6 2002:2003:1234:1::/64 2001:3001:1234:1::/64 log
```

4. Apply the ACL that you created to the appropriate interface.

> **NOTE**
> If an ACL with rules that contain the **log** keyword is applied to a management interface, logs are not recorded for that ACL.

5. (Optional) To display ACL logs, enter the **show access-list log buffer** command.

```
device# show access-list-log buffer
Frames Logged on interface 2/1 :
--------------------------------
Frame Received Time : Fri Dec 9 3:8:48 2011
Ethernet,       Src : (00:34:56:78:0a:ab), Dst: (00:12:ab:54:67:da)
  Ethtype           : 0x8100
  Vlan tag type     : 0x800
  VlanID            : 0x1
Internet proto, Src : 192.85.1.2, Dst: 192.0.0.1
  Interface         :
  Type of service   : 0
  Length            : 110
  Identification    : 0
  Fragmentation     : 00 00
  TTL               : 255
  protocol          : 253
  Checksum          : 39 3a
  Payload type      :
packet(s) repeated  : 30
Ingress Deny Logged
```

ACL Show and Clear commands

There is a full range of ACL show and clear commands. They are documented in the *Network OS Command Reference*, and listed here with descriptions.

TABLE 15 ACL Show commands in the Network OS Command Reference

Command	Description
show access-list	For a given network protocol and inbound/outbound direction, displays ACL information. You can show information for a specified ACL or only for that ACL on a specified interface or RBridge. You can also display information for all ACLs bound to a specified switch interface, RBridge, or VXLAN overlay gateway.
show access-list-log buffer	Displays the contents of the ACL buffer.
show access-list-log buffer config	Displays the ACL buffer configuration.
show running-config access-list	For a given network protocol and standard/extended type, displays ACL configuration. You can show the configuration of a specified ACL or for all such ACLs.
show statistics access-list	For a given network protocol and inbound/outbound direction, displays statistical information—for ACL rules that include the **count** keyword. You can show statistics for a specified ACL or only for that ACL on a specified interface or Rbridge. You can also display statistical information for all ACLs bound to a specified switch interface, RBridge, or VXLAN overlay gateway.

TABLE 16 ACL Clear commands in the Network OS Command Reference

Command	Description
clear counters access-list	For a given network protocol and inbound/outbound direction, clears ACL statistical information. You can clear all statistics for a specified ACL or only for that ACL on a specified interface or RBridge. You can also clear statistical information for all ACLs bound to a specified switch interface, RBridge, or VXLAN overlay gateway.

Configuring Dynamic ARP Inspection (DAI)

Dynamic ARP inspection (DAI) overview

Dynamic ARP inspection (DAI) is a security feature that validates address resolution protocol (ARP) packets in a subnet, and discards packets with invalid IP/MAC address bindings.

Address resolution protocol (ARP)

When forwarding traffic, a device needs to know the destination's MAC address, because each IP packet is encapsulated in a MAC packet. The MAC address is needed not only for the packet's final destination but also for a next hop towards the destination.

When the destination's IP address is known, to get the MAC address a device first searches its ARP cache. A match for the IP address supplies the corresponding MAC address. Otherwise, the device broadcasts an ARP request. The devices on the subnet receive such ARP requests, and the host whose IP address matches sends an ARP reply that includes its MAC address.

ARP poisoning

An ARP poisoning attack, also known as ARP spoofing, targets the ARP caches of devices connected to the subnet, with the goal of intercepting traffic.

A malicious host might use one of the following tactics:

- Send ARP packets claiming to have an IP address that actually belongs to another host.
- Reply to an ARP request with its own MAC address, thereby causing other hosts on the subnet to store this information in their ARP tables, even replacing an existing ARP entry.
- Send gratuitous replies without having received any ARP requests.

If the poisoning succeeds, traffic intended for the device under attack is instead routed to the attacker computer. The attacker has various options:

- Not forward any traffic to the computer under attack or forward some of the traffic, but not all of it (denial-of-service attacks).
- Forward inspected traffic to the compromised device (interception).
- Modify the traffic and then forward it (man-in-the-middle attack).

Dynamic ARP inspection (DAI)

On VLANs, dynamic ARP inspection (DAI) can examine incoming ARP packets. DAI discards packets with invalid IP/MAC address bindings, guarding against ARP-poisoning attacks; only valid ARP requests and responses are relayed.

Towards enabling DAI, you need to decide which ports you are defining as trusted and which as untrusted. ARP packets on trusted ports bypass all DAI validations and are forwarded as required. DAI examines ARP packets only on untrusted ports.

NOTE
DAI is currently supported only in non-DHCP environments. You specify valid, static IP/MAC address bindings in the **permit** statements of ARP ACLs.

When DAI is implemented on a VLAN, it monitors untrusted ports as follows:

- Intercepts ARP requests and responses
- Compares the IP/MAC address bindings with the **permit** statements in the ACL applied to the VLAN
- Drops invalid packets
- Forwards valid packets to the appropriate destination, with the option of generating log entries

Implementing ARP ACLs for DAI

Address-resolution protocol (ARP) access-lists (ACLs), applied to untrusted ports, permit only ARP packets with specified IP/MAC address-bindings. Such ACLs implement dynamic ARP inspection (DAI).

DAI configuration guidelines

Follow these guidelines when implementing ARP ACLs for dynamic ARP inspection (DAI).

- DAI is available on Layer 2 VLANs and Layer 3 virtual ethernets (VEs).

  NOTE
  In this topic, the term VLAN can also include VEs (where relevant).

- VLANs supported for DAI include:
  - 802.1Q VLANs
  - VLANs that cross multiple VCS clusters
  - Virtual fabric (VFAB) VLANs
  - VEs under virtual routing and forwarding (VRF). Both default and non-default VRFs are supported.
- DAI is not supported for management interfaces or for Layer 3 router interfaces.
- On a VLAN with DAI enabled, the following types of member ports are supported for DAI:
  - Physical interfaces (<N>-gigabit Ethernet)—in switchport mode
  - Port-channel interfaces (LAGs or vLAGs)—in switchport mode
- In a VCS VLAN with multiple RBridges, enabling DAI on the primary RBridge automatically implements it on all of the VLAN RBridges.
- On DAI-enabled VLANs, the rate limit per chip is 64 ARP packets per second (pps).

Creating an ARP-ACL

Use this procedure to create an ARP access-list (ACL) and **permit** rules.

1. Enter **configure terminal** to access global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter the **arp access-list** command to create the access list.

   ```
   device(config)# arp access-list arpACL1
   ```

3. For each ACL rule, enter a **permit ip host** command.

```
device(config-arp-acl)# permit ip host 1.1.1.1 mac host 0020.2222.2222
device(config-arp-acl)# permit ip host 1.1.1.2 mac host 0020.2222.2223
```

Applying an ARP ACL to a VLAN

Use this procedure to apply an ARP ACL to a VLAN.

NOTE
To replace an ARP ACL on a VLAN, there is no need to remove a previously applied ACL. The most recent ACL applied replaces any previous ACL.

1. Enter **configure terminal** to change to global configuration mode.

```
device# configure terminal
```

2. Enter the **interface vlan** command to access the VLAN.

```
device(conf)# interface vlan 200
```

3. Enter the **ip arp inspection filter** command, specifying the ACL.

```
device(conf-if-vlan-200)# ip arp inspection filter ARP_ACL_01
```

4. To return to global configuration mode—for example, to define trusted/untrusted interfaces and to enable DAI—enter **exit**.

```
device(conf-if-vlan-200)# exit
```

Defining trusted and untrusted interfaces under DAI

Use this procedure to specify untrusted and trusted interfaces under dynamic ARP inspection (DAI).

1. Enter **configure terminal** to change to global configuration mode.

```
device# configure terminal
```

2. For each interface that you need to define as trusted or untrusted, do the following:
   a) Enter the **interface** command to access interface configuration mode.

```
device(config)# interface tengigabitethernet 1/0/1
```

   b) To define the interface as trusted, enter the **ip arp inspection trust** command.

```
device(config-if-te-1/0/1)# ip arp inspection trust
```

   c) To redefine a currently trusted interface as untrusted (default), enter the **no ip arp inspection trust** command.

```
device(config-if-te-1/0/1)# no ip arp inspection trust
```

The following example defines a port-channel interface as trusted.

```
device# configure terminal
device(config)# interface port-channel 200
device(config-Port-channel-200)# ip arp inspection trust
```

Enabling and disabling dynamic ARP inspection (DAI)

Use this procedure to enable dynamic ARP inspection (DAI) on a VLAN.

1. Enter **configure terminal** to change to global configuration mode.

```
device# configure terminal
```

2. Enter the **interface vlan** command to access configuration mode on the VLAN for which you are enabling DAI.

```
device(config)# interface vlan 1001
```

3. To enable DAI, enter the **ip arp inspection** command.

```
device(config-if-vlan-1001)# ip arp inspection
```

4. To disable DAI, enter the **no ip arp inspection** command.

```
device(config-if-vlan-1001)# no ip arp inspection
```

Enabling and disabling DAI logging

Use this procedure to enable dynamic ARP inspection (DAI) on a VLAN, with terminal logging.

1. Enter **configure terminal** to access global configuration mode.

```
device# configure terminal
```

2. Enter the **arp access-list** command to create the access list.

```
device(config)# arp access-list arpACL1
```

3. For each ACL rule, enter a **permit ip host** command.

```
device(config-arp-acl)# permit ip host 1.1.1.1 mac host 0020.2222.2222 log
device(config-arp-acl)# permit ip host 1.1.1.2 mac host 0020.2222.2223
```

NOTE
If there is no **permit** statement that contains the **log** keyword, DAI logging does not occur.

4. Enter the **exit** command to return to global configuration mode.

```
device(config-arp-acl)# exit
```

5. Enter the **interface vlan** command to access configuration mode on the VLAN for which you are enabling DAI.

```
device(config)# interface vlan 200
```

6. Enter the **ip arp inspection filter** command, specifying an ACL containing a **permit** statement with the **log** keyword.

```
device(conf-if-vlan-200)# ip arp inspection filter arpACL1
```

7. Enter the ip **arp inspection acl-match matchlog** command.

```
device(conf-if-vlan-200)# ip arp inspection acl-match matchlog
```

8. Enter the **ip arp inspection** command to enable DAI.

```
device(config-if-vlan-200)# ip arp inspection
```

9. Enter the **end** command to return to privileged EXEC mode.

```
device(config-if-vlan-200)# end
```

10. Enter the terminal monitor command.

```
device# terminal monitor
```

The following example applies ARP_ACL_01 to VLAN 200, enables DAI logging on VLAN 200, enables DAI, and displays the log.

```
device# configure terminal
device(conf)# interface vlan 200
device(conf-if-vlan-200)# ip arp inspection filter ARP_ACL_01
device(conf-if-vlan-200)# ip arp inspection acl-match matchlog
device(conf-if-vlan-200)# ip arp inspection
device(conf-if-vlan-200)# end
device# terminal monitor
Terminal monitoring is enabled.
device# 015/03/11-18:47:06 PERMIT: arp Packet with srcIp=1.1.1.1, srcMac=0001.0001.0001,
dstIp=10.1.1.1,dstMac=ffff.ffff.ffff on Vlan: 11
2015/03/11-18:47:06 PERMIT: arp Packet with srcIp=1.1.1.1, srcMac=0001.0001.0001,
dstIp=10.1.1.1,dstMac=ffff.ffff.ffff on Vlan: 11
2015/03/11-18:47:06 PERMIT: arp Packet with srcIp=1.1.1.1, srcMac=0001.0001.0001,
dstIp=10.1.1.1,dstMac=ffff.ffff.ffff on Vlan: 11
2015/03/11-18:47:06 PERMIT: arp Packet with srcIp=1.1.1.1, srcMac=0001.0001.0001,
dstIp=10.1.1.1,dstMac=ffff.ffff.ffff on Vlan: 11
2015/03/11-18:47:06 PERMIT: arp Packet with srcIp=1.1.1.1, srcMac=0001.0001.0001,
dstIp=10.1.1.1,dstMac=ffff.ffff.ffff on Vlan: 11
2015/03/11-18:47:06 PERMIT: arp Packet with srcIp=1.1.1.1, srcMac=0001.0001.0001,
dstIp=10.1.1.1,dstMac=ffff.ffff.ffff on Vlan: 11
2015/03/11-18:47:06 PERMIT: arp Packet with srcIp=1.1.1.1, srcMac=0001.0001.0001,
dstIp=10.1.1.1,dstMac=ffff.ffff.ffff on Vlan: 11
2015/03/11-18:47:06 PERMIT: arp Packet with srcIp=1.1.1.1, srcMac=0001.0001.0001,
dstIp=10.1.1.1,dstMac=ffff.ffff.ffff on Vlan: 11
2015/03/11-18:47:06 PERMIT: arp Packet with srcIp=1.1.1.1, srcMac=0001.0001.0001,
dstIp=10.1.1.1,dstMac=ffff.ffff.ffff on Vlan: 11
```

DAI Show/Clear commands

There is a full range of dynamic ARP inspection (DAI) show and clear commands. They are documented in the *Network OS Command Reference*, and listed here with descriptions.

TABLE 17 DAI Show commands in the Network OS Command Reference

Command	Description
show arp access-list	For one or all ARP ACLs defined on a device, displays ACL names and their **permit** statements.
show ip arp inspection interfaces	For VLANs enabled for DAI, displays a list of trusted interfaces.
show ip arp inspection statistics	Displays DAI statistics for one or more DAI-enabled VLANs.
show ip arp inspection	Displays DAI information for one or more VLANs.

TABLE 18 DAI Clear commands in the Network OS Command Reference

Command	Description
clear ip arp inspection statistics	Clears DAI statistics for all DAI-enabled VLANs.

Configuring Fabric Authentication

Fabric authentication overview

When you connect a Brocade VCS Fabric to a Fabric OS fabric, the Fibre Channel E_Ports on the hardware connect through Inter-Switch Links (ISLs) to EX_Ports on an FC router, which in turn connects to the Fabric OS network.

Refer to the Fibre Channel ports overview section of the *Network OS Administrator's Guide*.

To ensure that no unauthorized devices can access the fabric, the software provides support for security policies and protocols capable of authenticating the software (E_Ports) to the EX_Ports on the FC router (FCR) that provides access to the SAN storage and services.

Understanding fabric authentication

This section presents a brief overview of SSH server key exchange, configuring an authentication policy and device authentication, and configuring SCC policy sets.

DH-CHAP

The software uses the Diffie-Hellman Challenge Handshake Authentication Protocol (DH-CHAP) to control access between devices. DH-CHAP is a password-based, key exchange authentication protocol that negotiates hash algorithms and Diffie Hellman (DH) groups before performing authentication. It supports both MD5 and SHA-1 hash algorithm-based authentication.

The Fibre Channel Security Protocol (FC-SP) defines the DH groups supported in the DH-CHAP protocol. Following current FC-SP standards, the software supports the following DH groups:

- 00 - DH Null option
- 01 - 1024 bit key
- 02 - 1280 bit key
- 03 - 1536 bit key
- 04 - 2048 bit key

To configure DH-CHAP authentication between devices (E_Ports) and FC routers (EX_Ports) you must apply a matching configuration to both sides of the connection. Each device must be configured locally.

Configuring an authentication policy

The device authentication (AUTH) policy initiates DH-CHAP authentication on all E_Ports. This policy is persistent across reboots, which means authentication will be initiated automatically on ports or devices brought online if the policy is active. You must configure the AUTH policy on all connected fabric entities.

If you are in logical chassis cluster mode, the authentication policy is *not* distributed across the cluster. The RBridge ID of the node should be used to configure protocol and policy configurations.

By default the policy is set to PASSIVE and you can change the policy. All changes to the AUTH policy take effect during the next authentication request. This includes starting authentication on all E_Ports on the local device if the policy is changed to ON or ACTIVE, and clearing the authentication requirement if the policy is changed to OFF.

Authentication policy configuration is not distributed across the cluster. The RBridge ID of the node should be used to configure protocol and policy configurations.

You can set the authentication policy to any of the values listed in the following table. The remaining attributes are optional.

TABLE 19 User account attributes

Setting	Description
ON	Strict authentication is enforced on all E_Ports. During device initialization, authentication is initiated on all E_Ports automatically. The authentication handshaking is completed before the devices exchange the fabric parameters (EFP) for E_Port bring-up. If the connecting device does not support the authentication or the policy is turned off, all ports are disabled and the ISL goes down.
ACTIVE	A device with an ACTIVE policy is more tolerant and can connect to a device with any type of policy. During device initialization, authentication is initiated on all E_Ports, but the port is not disabled if the connecting device does not support authentication, or if the authentication policy is turned off.
PASSIVE (default)	The device does not initiate authentication, but participates in authentication if the connecting device initiates authentication. The device does not start authentication on E_Ports, but accepts the incoming authentication requests, and will not be disabled if the connecting device does not support authentication or the policy is turned off.
OFF	The device does not support authentication, and rejects any authentication negotiation request from a neighbor device or device. A device with the policy set to OFF should not be connected to a device with a policy set to ON. A policy set to ON policy is strict and disables the port if a peer rejects the authentication. DH CHAP shared secrets must be configured on both sides of the connection before you can change the policy from an OFF state to an ON state.

The behavior of the policy between two adjacent devices is defined as follows:

- If the policy is ON or ACTIVE, the device sends an Authentication Negotiation request to the connecting device.
- If the connecting device does not support authentication or the policy is OFF, the request is rejected.
- Once the authentication negotiation succeeds, the DH-CHAP authentication is initiated. If DH-CHAP authentication fails, the port is disabled, regardless of the policy settings.

The policy defines the responses of the host if the connecting device does not support authentication. By default, the policy is set to PASSIVE and you can change the policy with the **fcsp auth** command. This includes starting authentication on all E_Ports if the policy is changed to ON or ACTIVE, and clearing the authentication if the policy is changed to OFF. Before enabling the policy, you must install the DH-CHAP shared secrets. Refer to Configuring DH-CHAP shared secrets on page 112.

Configuring DH-CHAP shared secrets

To configure the DH-CHAP shared secrets, enter the **fcsp auth-secret** command in privileged EXEC mode. Provide the following information as shown in the example:

- The world wide name (WWN) of the peer.
- The secret of the peer that authenticates the peer to the local switch.

- The local secret that authenticates the local switch to the peer.

  NOTE
  Only the following non-alphanumeric characters are valid for the secret key:@ $ % ^ & * ( ) _ + - < > { } [ ] ; ' :

```
switch# fcsp auth-secret dh-chap node 10:00:00:05:1e:7a:c3:00 peer-secret    12345678 local-secret 87654321
Shared secret is configured successfully.
```

To display the device (WWN) for which the shared secret is configured, use the **show fcsp auth-secret dh-chap** command in privileged EXEC mode.

```
switch# show fcsp auth-secret dh-chap 10:00:00:05:1e:7a:c3:00
```

To remove the shared secrets, use the **no fcsp auth-secret** command in privileged EXEC mode.

```
switch# no fcsp auth-secret dh-chap node 10:00:00:05:1e:7a:c3:00
Shared secret successfully removed
```

Setting up secret keys

Setting up secret keys can quickly become an administrative challenge as your fabric size increases. As a minimum, key pairs need to be installed on all connected fabric entities. However, when connections change, you must install new key pairs to accommodate these changes. If you anticipate this situation, you may install key pairs for all possible connections up front, thus enabling links to change arbitrarily while still maintaining a valid key pair for any new connection.

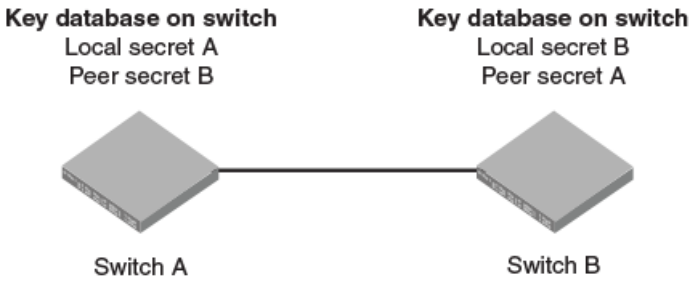
Shared secret keys

When you configure device ports for DH-CHAP authentication, you define a pair of shared secrets known to both devices as a secret key pair. A key pair consists of a local secret and a peer secret. The local secret uniquely identifies the local device. The peer secret uniquely identifies the entity to which the local device may authenticate. Every device may share a secret key pair with any other device or host in a fabric.

Shared secret keys have the following characteristics:

- The shared secrets must be configured locally on every device.
- If shared secrets are not set up for a link, authentication fails. The “Authentication Failed” error is reported for the port.
- The minimum length of a shared secret is 8 bytes and the maximum 40 bytes.

FIGURE 2 DH-CHAP authentication



The preceding figure illustrates how the secrets are configured. Assume two devices, A and B. Each device has a local secret (local secret A and local secret B), and a matching peer secret (peer secret A and peer secret B). If device B wants to shake hands with A, it will use A’s local secret (B’s peer secret A) to send the information. In doing so, A authenticates B by confirming its identity through the exchange of matching secret pairs. Conversely, B authenticates A when A sends information to B using B’s local secret (A’s peer secret B).

On the FC router, the authentication configuration for EX_Ports is set to fixed default values and cannot be changed. The Fabric OS **authutil** command is applicable only to the E_Ports on the FC router, not to EX_Ports. The following table shows the default authentication configuration for EX_Ports:

TABLE 20 Default EX_Port configuration

Operand	Value
Auth-type	DHCHAP
Auth-Policy	PASSIVE
Auth-Group	* (0, 1, 2, 3, 4)
Auth-Hash	msd5, sha1

Setting the authentication policy parameters

The following procedure configures an authentication policy auth-type DH-CHAP (only option), a DH group of 2, and a hash type of md5. The switch policy is set to OFF until you are ready to explicitly activate the policy.

1. In privileged EXEC mode, use the **configure terminal** command to enter global configuration mode.

```
switch# configure terminal
Entering configuration mode terminal
```

2. Enter the **fcsp auth** command with the specified parameters.

```
switch(config)# fcsp auth auth-type dh-chap hash md5 group 2 switch policy off
```

3. Enter the **do show running-config fcsp auth** command to verify the configuration.

```
switch(config)# do show running-config fcsp auth
fcsp auth group 2
fcsp auth hash md5
fcsp auth policy switch off
```

Activating the authentication policy

1. In privileged EXEC mode, issue the **configure terminal** command to enter global configuration mode.

```
switch# configure terminal
Entering configuration mode terminal
```

2. Enter the **fcsp auth auth-type** command to change the policy state from OFF to ON.

```
switch(config)# fcsp auth auth-type switch policy on
```

3. Enter the **do show running-config fcsp auth** command to verify the configuration.

```
switch(config)# do show running-config fcsp auth
fcsp auth group 2
fcsp auth hash md5
fcsp auth policy switch on
```

Switch connection control policy

The Switch Connection Control (SCC) policy controls access between neighboring devices. The policy defines and restricts which devices can join the fabric. Each time an E_Port-to-EX_Port connection is attempted, the devices are checked against the policy and the connection is either accepted or rejected depending on whether the connecting device is listed in the policy. The policy is named SCC_POLICY and accepts members listed as world wide names (WWNs).

A device configured with an active SCC policy reviews its database whenever a neighboring device tries to establish a connection. If the WWN of the connecting device is found in the SCC active policy database, the connecting device is allowed to join the fabric. If the neighboring device is not specified in the SCC policy active list, both devices are segmented.

By default, any device is allowed to join the fabric; the SCC policy is not enforced until it is created and activated. Creating a policy without any entries blocks access from all devices. The local switch is not required to be included in a switch-local SCC policy.

SCC policy commands are not distributed across the cluster. The RBridge ID of the node should be used to configure policy configurations.

Configuring defined and active SCC policy sets

The Switch Connection Control (SCC) policy maintains two versions, active, and defined, and creating a policy includes two distinct operations:

1. Creating the defined SCC policy set.

2. Activating the SCC policy.

The defined policy includes a list of WWN members and it is configurable. You can create the SCC policy and its members using a single command, **secpolicy defined-policy SCC_POLICY**. Or you can create the SCC policy first and add the members later. You can modify the defined policy at any time thereafter.

When you create the SCC policy and its defined member set, it remains inactive until you explicitly activate the policy with the **secpolicy activate** command. The SCC policy is enforced on the E_Ports only after you activate the policy. When the policy is active, only the members included in the activated policy can communicate with each other. If you add additional devices to the defined policy, they remain inactive and access is blocked until you active the defined policy again.

Follow these guidelines and restrictions when configuring SCC policy:

- During the configuration replay operation, the defined and active policies are replayed and the E_Ports are enabled or disabled based on the SCC policy entries in the active policy list.

  During a configuration replay operation, if an E_Port is already disabled due to a violation, it will not come online even if the WWN entry is found in the active policy list. You must explicitly bring up the E_Port to enforce the active policy.
- During execution of the **copy file running-config** command, only the defined policy in the switch is updated with the config file entries; the active policy entries remain unchanged. In this case, you must use the **secpolicy activate** command to activate the defined policy list.
- If an empty policy is created and activated, but not saved, all Fibre Channel (FC) E_Ports will be in the disabled state after a reboot.
- Network OS requires that you invoke the **shutdown** command, followed by the **no shutdown** command to bring up the E_Port. Invoking the **no shutdown** command alone does not enable the port.

Creating a defined SCC policy

The following procedure creates a Switch Connection Control (SCC) policy, adds two members, and verifies the configuration.

1. In privileged EXEC mode, issue the **configure terminal** command to enter global configuration mode.

2. Enter the **secpolicy defined-policy SCC_POLICY** command.

   This command places you into the defined SCC configuration mode where you can add policy member WWNs.
3. Specify a policy member with the **member-entry** *WWN* command.
4. Specify a second policy member with the **member-entry** *WWN* command.
5. Exit the defined SCC configuration mode.
6. Enter the **do show running-config secpolicy defined-policy** command to verify the configuration.

Creating an SCC policy in VCS mode

This example creates an SCC policy in VCS mode:

```
switch# config
Entering configuration mode terminal

switch(config)# rbridge-id 3

switch(config-rbridge-id-3)# secpolicy defined-policy SCC_POLICY

switch(config-defined-policy-SCC_POLICY)# exit

switch(config)#
```

Creating an SCC policy and adding members into the defined policy set in VCS mode

This VCS mode example creates a SCC policy and adds members into the defined policy set

```
switch# config
Entering configuration mode terminal

switch(config)# rbridge-id 3

switch(config-rbridge-id-3)# secpolicy defined-policy SCC_POLICY member-entry 10:00:00:05:1e:00:69:00

switch(config-member-entry-10:00:00:05:1e:00:69:00)# exit

switch(config-defined-policy-SCC_POLICY)# exit

switch(config-rbridge-id-3)# exit
```

Modifying the SCC policy

The same command sequence that creates the Switch Connection Control (SCC) policy adds additional members. The defined SCC member entries are cumulative. Use the **no member-entry** command to remove members from the policy.

The following example adds a member and subsequently removes the same added member:

VCS mode example

```
switch# configure terminal
Entering configuration mode terminal

switch(config)# rbridge-id 3

switch(config-rbridge-id-3)# secpolicy defined-policy SCC_POLICY

switch(config-defined-policy-SCC_POLICY)# member-entry 10:00:00:05:1e:00:69:00

switch(config-defined-policy-SCC_POLICY)# no member-entry 10:00:00:05:1e:00:69:00

switch(config-defined-policy-SCC_POLICY)# exit

switch(config)# do show running-config secpolicy defined-policy
```

```
secpolicy defined-policy SCC_POLICY
 member-entry 10:00:00:05:1e:00:69:00
 !
 member-entry 10:00:00:08:2f:00:79:00
```

Activating the SCC policy

1. Define the SCC policy as shown in section Creating a defined SCC policy on page 115.
2. Enter the **secpolicy activate** command in privileged EXEC mode.
3. Enter the **do show running-config secpolicy active -policy** command to verify the configuration.

VCS mode example

```
switch# secpolicy activate rbridge-id 3

switch# do show running-config rbridge-id 3 secpolicy defined-policy rbridge-id 3
secpolicy defined-policy SCC_POLICY
 member-entry aa:aa:aa:aa:aa:aa:aa:aa
 !
 member-entry bb:bb:bb:bb:bb:bb:bb:bb
 !
 member-entry cc:cc:cc:cc:cc:cc:cc:cc
 !
!
```

Removing the SCC Policy

1. In privileged EXEC mode, issue the **configure terminal** command to enter global configuration mode.
2. Enter the **no secpolicy defined-policy SCC_POLICY** command.
3. Exit global configuration mode.
4. Activate the SCC policy to save the defined policy configuration to the active configuration.
5. Enter the **do show running-config secpolicy active-policy** command to verify that the policy is empty.

Removing an entry from the policy list of RBridge ID 3 in VCS mode

```
switch# configure terminal
Entering configuration mode terminal

switch(config)# rbridge-id 3

switch(config-rbridge-id-3)# no secpolicy defined-policy SCC_POLICY member-entry 10:00:00:05:1e:00:69:01

switch(config)# exit

switch# do show running-config secpolicy active-policy
% No entries found.
```

Removing the SCC_POLICY entry of RBridge ID 3 in VCS mode

```
switch# configure terminal
Entering configuration mode terminal

switch(config)# rbridge-id 3

switch(config-rbridge-id-3)# no secpolicy defined-policy SCC_POLICY

switch(config)# exit

switch# do show running-config secpolicy active-policy
% No entries found.
```

Port Security

Port security can be used to prevent administrators or malicious users from being able to change the MAC address of a virtual machine (VM) in a data center environment. This is especially helpful in virtual desktop infrastructure (VDI) environments, where users might have full administrative control of the VM and can change the MAC address of a virtual network interface card (vNIC). Here port security can be used to provide more control over the behavior of VMs.

Port security can be used to prevent administrators or malicious users from being able to change the MAC address of a virtual machine (VM) in a data center environment. This is especially helpful in virtual desktop infrastructure (VDI) environments, where users might have full administrative control of the VM and can change the MAC address of a virtual network interface card (vNIC). Here port security can be used to provide more control over the behavior of VMs. The secured ports can be categorized as either trusted or untrusted. The administrator can apply policies appropriate to those categories to protect against various types of attacks. Port security features can be turned on to obtain the most robust port-security level that is appropriate. Basic port-security features are enabled in the device's default configuration. Additional features can be enabled with minimal configuration steps.

The following MAC port-security features enhance security at Layer 2:

- MAC address limiting—This restricts input to an interface by limiting and identifying the MAC addresses of workstations that are allowed to access the port. When secure MAC addresses are assigned to a secure port, the port does not forward packets with source addresses outside the group of defined addresses.
- OUI-based port security—If an administrator knows which types of systems are connecting to the network, it is possible to configure an Organizationally Unique Identifier (OUI) on a secure port to ensure that only traffic coming from devices that are part of the configured OUI is forwarded.
- Port security with sticky MAC addresses—Using sticky MAC addresses is similar to using static secure MAC addresses, but sticky MAC addresses are learned dynamically. These addresses are retained when a link goes down.

Default port security configuration options

Port security is disabled by default. The following table summarizes default port-security configuration options that are applied to an interface when it is made a secure port.

TABLE 21 Default configurations for port security

Feature	Default configuration
Max. number of secure MAC addresses	8192
Violation mode	Shutdown
Shutdown time (minutes)	0

Port security commands

Port security is enabled on an interface by means of a series of switchport commands. For configuration examples, refer to Configuring port security section and the *Network OS Command Reference*.

Command	Description
switchport port-security	Enables or disables port security on an interface port.
switchport port-security mac-address	Configures the MAC address option for port security on an interface port.
switchport port-security max	Configures the maximum number of MAC addresses used for port security on an interface port.
switchport port-security oui	Configures an Organizationally Unique Identifier (OUI) MAC address for port security on an interface port.
switchport port-security shutdown-time	Configures the shutdown-time option for port security on an interface port.
switchport port-security sticky	Converts dynamic MAC addresses to sticky secure MAC addresses.
switchport port-security violation	Configures the violation response options for port security on an interface.

Port security troubleshooting commands

The following commands can be used to troubleshoot port security configuration issues.

Command	Description
show ip interface brief	Displays port-security status when the port-security feature is applied.
show port-security	Displays the configuration information related to port-security.
show port-security addresses	Displays the configuration information related to port-security addresses.
show port-security interface	Displays the configuration information related to port-security interfaces.
show port-security oui interface	Displays the configuration information related to port-security for Organizationally Unique Identifier (OUI) interfaces.
show port-security sticky interface	Displays the configuration information related to port-security for a sticky interface

Port security guidelines and restrictions

Note the following guidelines and restrictions for configuring port security:

- A port mode change is not allowed when port security is enabled on the interface.
- A maximum of 4 OUIs are allowed per secure port. A maximum of 20 secure ports are allowed to enable OUI-based port security.
- Static secure MAC addresses are not supported for OUI-based port security.
- When the user tries to configure the first OUI IPv4 address on a secure port, traffic is flooded until all entries are programmed in the hardware.
- If a port-security-based change occurs when a port is shut down, the shutdown timer is not triggered. Consequently, the user must restore the full functionality of the port.
- When port security causes a port to be shut down and the user manually changes the shutdown time, the shutdown timer is reset and the timer starts with the new shutdown time.
- A secure port cannot be a destination port for Switch Port Analyzer (SPAN) purposes, because the port cannot be a Layer 2 port.

- Port security configurations are not allowed on member interfaces of a link aggregation group (LAG). They are allowed on the LAG interface, however, as they are in other Layer 2 configurations.
- Static MAC addresses cannot be configured on a secure port. They must be configured as secure MAC addresses on the secure port.
- Access control lists (ACLs) take precedence over the port security feature.

Configuring port security

The following section covers how to configure port security for access and trunk ports, set port-security MAC address limits and shutdown time, set up OUI-based port security, and configure port security with sticky MAC addresses.

Refer also to Port Security on page 118.

Configuring port security on an access port

To enable port security on an access port, do the following in global configuration mode.

1. Enable interface subconfiguration mode for the interface you want to modify.

```
switch(config)# interface TenGigabitEthernet 1/0
```

2. Put the interface in Layer 2 mode by using the **switchport** command.

```
switch(conf-if-te-1/0)# switchport
```

3. Enable switchport security by using the **switchport port-security** command.

```
switch(conf-if-te-1/0)# switchport port-security
```

Configuring port security on a trunk port

To enable port security on a trunk port, do the following in global configuration mode.

1. Enable interface subconfiguration mode for the interface you want to modify.

```
switch(config)# interface TenGigabitEthernet 1/0
```

2. Put the interface in Layer 2 mode by using the **switchport** command.

```
switch(conf-if-te-1/0)# switchport
```

3. Set the mode of the interface to trunk.

```
switch(conf-if-te-1/0)# switchport mode trunk
```

4. Set the VLANs that will transmit and receive through the Layer 2 interface.

```
switch(conf-if-te-1/0)# switchport trunk allowed vlan add 100
```

5. Enable switchport security by using the **switchport port-security** command.

```
switch(conf-if-te-1/0)# switchport port-security
```

Configuring port-security MAC address limits

To configure the MAC address option for port security on an interface port, do the following in global configuration mode.

1. Enable interface subconfiguration mode for the interface you want to modify.

```
switch(config)# interface TenGigabitEthernet 1/0
```

2. Put the interface in Layer 2 mode by using the **switchport** command.

```
switch(conf-if-te-1/0)# switchport
```

3. Set the MAC address and VLAN ID for the interface.

```
switch(conf-if-te-1/0)# switchport port-security mac-address 1000.2000.3000 vlan 100
```

Configuring port-security shutdown time

You can configure two responses to a violation of port security: **restrict** and **shutdown**.

- The **restrict** option drops packets that have unknown source addresses until you remove a sufficient number of secure MAC addresses until this value is below that set by the **switchport port-security max** command.
- The **shutdown** option puts the interface in the error-disabled state immediately for a predetermined amount of time.

To configure the port-security shutdown time for an interface port, do the following in global configuration mode.

1. Enable interface subconfiguration mode for the interface you want to modify.

```
switch(config)# interface TenGigabitEthernet 1/0
```

2. Put the interface in Layer 2 mode by using the **switchport** command.

```
switch(conf-if-te-1/0)# switchport
```

3. Set the violation response option to shutdown.

```
switch(conf-if-te-1/0)# switchport port-security violation shutdown
```

4. Set the shutdown time, in minutes.

```
switch(conf-if-te-1/0)# switchport port-security shutdown-time 10
```

Configuring OUI-based port security

If you know which types of systems are connected to your network, use this security feature to configure an Organizationally Unique Identifier (OUI) MAC address on a secure port. This ensures that only traffic from a known OUI MAC address is forwarded.

To configure OUI-based port security, do the following in global configuration mode.

1. Enable interface subconfiguration mode for the interface you want to modify.

```
switch(config)# interface TenGigabitEthernet 1/0
```

2. Put the interface in Layer 2 mode by using the **switchport** command.

```
switch(conf-if-te-1/0)# switchport
```

3. Configure a permitted OUI MAC address by using the **switchport port-security oui** command.

```
switch(conf-if-te-1/0)# switchport port-security oui 2000.3000.4000
```

Configuring port security with sticky MAC addresses

You can configure an interface to convert dynamic MAC addresses to sticky secure MAC addresses and add them to the running-config by enabling sticky learning. This converts all dynamic secure MAC addresses, including those learned dynamically before sticky learning was enabled, to sticky secure MAC addresses.

To configure sticky MAC addresses on a secure port, do the following in global configuration mode.

1. Enable interface subconfiguration mode for the interface you want to modify.

```
switch(config)# interface TenGigabitEthernet 1/0
```

2. Put the interface in Layer 2 mode by using the **switchport** command.

```
switch(conf-if-te-1/0)# switchport
```

3. Enable switchport security by using the **switchport port-security oui** command.

```
switch(conf-if-te-1/0)# switchport port-security oui 2000.3000.4000
```

4. Configure the sticky option.

```
switch(conf-if-te-1/0)# switchport port-security sticky
```

Configuring SSH

Configuring SSH encryption protocol

Secure Shell (SSH) is a protocol which encrypts remote access connections to network devices.

Using encrypted shared keys, SSH authenticates clients or servers, ensuring that the devices accessing your network are authentic.

The steps to configuring SSH are:

- Configure the SSH Server and Client ciphers.
- Configure the SSH Server and Client key-exchange algorithms.
- Configure the SSH Server and Client MACs.

Ciphers, non-CBC ciphers, algorithms, and MACs are not mutually exclusive. Any combination can be configured on the device.

Configuring SSH ciphers

Configures the Secure Shell (SSH) ciphers.

Refer to the online help on the device for the complete list of supported ciphers.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter Rbridge-ID configuration mode.

```
device(config)# rbridge-id 1
```

3. Use the **ssh server cipher** command to set the server cipher for SSH.

   You can use multiple ciphers by separating the string names with commas.

```
device(config-rbridge-id-1)# ssh server cipher aes192-cbc,aes128-ctr
```

4. Use the **ssh client cipher** command to set the client cipher for SSH.

   You can use multiple ciphers by separating the string names with commas.

```
device(config-rbridge-id-1)# ssh client cipher aes192-cbc,aes128-ctr
```

5. Restart the SSH server using the **no ssh server shutdown** command.

6. Confirm the cipher setting with the **show running-config** command or the **show ssh** command.

```
device(config-rbridge-id-1)## show running-config rbridge-id ssh server cipher
rbridge-id 1
ssh server cipher aes192-cbc,aes128-ctr

device(config-rbridge-id-1)## show running-config rbridge-id ssh client cipher
rbridge-id 1
ssh client cipher aes192-cbc,aes128-ctr

device(config-rbridge-id-1)# do show ssh server status rbridge-id 1
rbridge-id 1:SSH server status:Enabled
rbridge-id 1: SSH Server Cipher: aes192-cbc,aes128-ctr

device(config-rbridge-id-176)# do show ssh client status rbridge-id 1
rbridge-id 1: SSH Client Cipher: aes192-cbc, aes128-ctr
```

Typical command sequence:

```
device# configure terminal
device(config)# rbridge-id 1
device(config-rbridge-id-1)# ssh server cipher aes192-cbc, aes128-ctr
device(config-rbridge-id-1)# ssh client cipher aes192-cbc, aes128-ctr
device(config-rbridge-id-1)# do show ssh server status rbridge-id 1
rbridge-id 1:SSH server status:Enabled
rbridge-id 1: SSH Server Cipher: aes192-cbc,aes128-ctr

device(config-rbridge-id-176)# do show ssh client status rbridge-id 1
rbridge-id 1: SSH Client Cipher: aes192-cbc, aes128-ctr
```

Configuring non-CBC SSH cipher

supports Cipher Block Chaining (CBC) ciphers and non-CBC ciphers. This task configures the non-CBC ciphers for Secure Shell (SSH).

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter Rbridge-ID configuration mode.

```
device(config)# rbridge-id 1
```

3. Use the **ssh server cipher** command to set the server cipher for SSH.

```
device(config-rbridge-id-1)# ssh server cipher non-cbc
```

4. Use the **ssh client cipher** command to set the client cipher for SSH.

```
device(config-rbridge-id-1)# ssh client cipher non-cbc
```

5. Restart the SSH server using the **no ssh server shutdown** command.
6. Confirm the cipher setting with the **show running-config** command to set the client cipher version for SSH.

```
device(config-rbridge-id-1)# ssh client cipher non-cbc
```

Typical command sequence:

```
device# configure terminal
device(config)# rbridge-id 1
device(config-rbridge-id-1)# ssh server cipher non-cbc
device(config-rbridge-id-1)# ssh client cipher non-cbc
```

Removing an SSH cipher

The "no" form of the **ssh server cipher** and **ssh client cipher** commands sets the SSH ciphers back to the default algorithms.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter Rbridge-ID configuration mode.

```
device(config)# rbridge-id 1
```

3. Use the **ssh server cipher** command to remove the server cipher for SSH.

   You can remove multiple ciphers by separating the string names with commas.

```
device(config-rbridge-id-1)# no ssh server cipher
```

4. Use the **ssh client cipher** command to remove the client cipher for SSH.

   You can remove multiple ciphers by separating the string names with commas.

```
device(config-rbridge-id-1)# no ssh client cipher
```

Configuring SSH key-exchange

The SSH key-exchange specifies the algorithms used for generating one-time session keys for encryption and authentication with the SSH server.

Refer to the online help on the device for the complete list of supported key exchange algorithms.

For backward compatibility, the string "dh-group-14" is also acceptable in place of "diffie-hellmangroup14-sha1".

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter Rbridge-ID configuration mode.

```
device(config)# rbridge-id 3
```

3. Use the **ssh server key-exchange** command to set the key exchange algorithm for the server.

   You can use multiple key exchange algorithms by separating the string names with commas.

```
device(config-rbridge-id-3)# ssh server key-exchange diffie-hellman-group14-sha1,ecdh-sha2-nistp521
```

4. Use the **ssh client key-exchange** command to set the key exchange algorithm for the client.

   You can use multiple key exchange algorithms by separating the string names with commas.

```
device(config-rbridge-id-3)# ssh client key-exchange diffie-hellman-group14-sha1,ecdh-sha2-nistp521
```

5. Restart the SSH server using the **no ssh server shutdown** command.

Typical command sequence example.

```
device# configure terminal
device(config)# rbridge-id 3
device(config-rbridge-id-3)# ssh server key-exchange diffie-hellman-group14-sha1, ecdh-sha2-nistp521
device(config-rbridge-id-3)# ssh client key-exchange diffie-hellman-group14-sha1, ecdh-sha2-nistp521
```

Removing an SSH key-exchange

The "no" version of the **ssh server key-exchange** command is used to reset the SSH key exchange algorithms back to the default values.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter Rbridge-ID configuration mode.

```
device(config)# rbridge-id 3
```

3. Use the **ssh server** command to reset the key exchange algorithm for the server to the default value.

```
device(config-rbridge-id-3)# no ssh server key-exchange
```

4. Use the **ssh client** command to reset the key exchange algorithm for the client to the default value.

```
device(config-rbridge-id-3)# no ssh client key-exchange
```

Configuring SSH MAC

Configures SSH Server and Client Message Authentication Codes (MACs).

SSH server must be enabled.

Refer to the online help on the device for the complete list of supported MACs.

1. Enter configure terminal mode.

```
switch#configure terminal
```

2. Enter RBridge ID mode.

```
switch(config)#rbridge-id 176
```

3. On the SSH server, enter the **ssh server mac** command to configure the SSH server information.
   You can use multiple MACs by separating the string names with commas.

```
switch(config-rbridge-id-176)# ssh server mac
```

4. On the SSH client, enter the **ssh client mac** command to configure the SSH client information.
   You can use multiple MACs by separating the string names with commas.

```
switch(config-rbridge-id-176)# ssh client mac
```

5. Restart the SSH server using the **no ssh server shutdown** command.

6. Enter the **show running-config** command or the the **show ssh** command to confirm the SSH configuration information.

```
switch(config-rbridge-id-176)# do show running-config rbridge-id ssh server
rbridge-id 176
ssh server mac hmac-sha1,hmac-sha2-256,hmac-sha2-512
ssh server key rsa 2048
ssh server key ecdsa 256
ssh server key dsa

switch(config-rbridge-id-176)# do show running-config rbridge-id ssh client
rbridge-id 176
ssh client mac hmac-sha1,hmac-sha2-256,hmac-sha2-512

device(config-rbridge-id-1)# do show ssh server status rbridge-id 1
rbridge-id 1:SSH server status:Enabled
rbridge-id 1:SSH server Mac: hmac-sha1,hmac-sha2-256,hmac-sha2-512

switch(config-rbridge-id-176)# do show ssh client status rbridge-id 176
rbridge-id 176:SSH Client Mac: hmac-sha1,hmac-sha2-256,hmac-sha2-512
```

Removing an SSH MAC

The "no" form of the **ssh server mac** and **ssh client mac** commands removes the MACS.

1. Enter configure terminal mode.

```
switch#configure terminal
```

2. Enter RBridge ID mode.

```
switch(config)#rbridge-id 176
```

3. On the SSH server, enter the **ssh server mac** command to set the SSH server MACs to default values.

```
switch(config-rbridge-id-176)# no ssh server mac
```

4. Restart the SSH server using the **no ssh server shutdown** command.
5. On the SSH client, enter the **ssh client mac** command to set the SSH server MACs to default values.

```
switch(config-rbridge-id-176)# no ssh client mac
```

Router Advertisement (RA) Guard

RA Guard overview

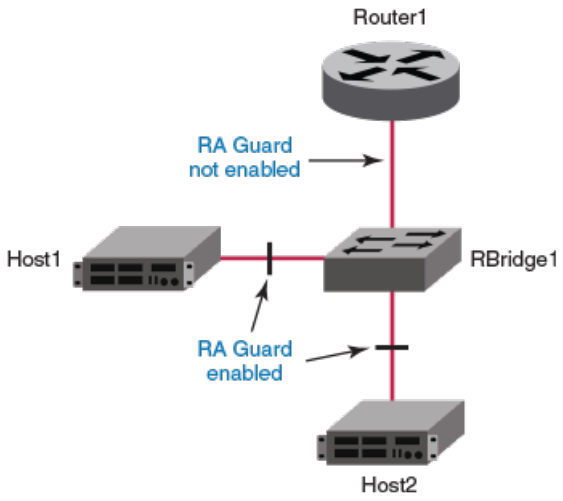
In a routed network, devices are configured to send router advertisements (RAs). RAs enable link nodes to discover routers, allowing the nodes to autoconfigure.

However, routed protocols are susceptible to rogue RAs generated by unauthorized or improperly configured devices connected to the segment. RA Guard prevents RAs from such devices from entering an L2 network.

RA Guard is effective in an environment where messages between IPv6 end-devices traverse L2 networking devices.

In the following diagram, the system is configured to block RAs on ports connected to hosts and to allow RAs on router-facing ports.

FIGURE 3 RA Guard scenario



RA Guard configuration guidelines

When implementing RA Guard, be aware of these configuration guidelines.

If RA Guard is enabled on an interface, this defines an internal ACL rule, for example:

```
seq 10 hard-drop IPv6-ICMP any any icmp-type 134 icmp-code 0
```

Be aware of the following ACL-related issues:

- RA Guard requires a profile with Ternary Content-Addressable Memory (TCAM) resources for IPv6 ACLs. Such resources are shared by RA Guard and user-defined ACLs.
- An RA Guard rule takes precedence over user-configured ACL rules applied to that interface.

NOTE
For more information on ACLs, refer to ACLs on page 83.

You can apply RA Guard only on the Physical Switchport interface type.

RA Guard is only supported on the Brocade VDX 6740 and the Brocade VDX 6940.

Enabling and disabling RA Guard

Use this procedure to enable or disable RA Guard on an interface.

1. Enter **configure** to change to global configuration mode.

```
switch# configure
```

2. Enter the **interface** command, specifying the interface type and the rbridge-id/slot/port number.

```
switch(config)# interface ten 122/5/22
```

3. To enable RA Guard on this interface, enter **ipv6 raguard**.

```
switch(conf-if-te-122/5/22)# ipv6 raguard
```

4. To disable RA Guard on this interface, enter **no ipv6 raguard**.

```
switch(conf-if-te-122/5/22)# no ipv6 raguard
```

RA Guard Show commands

There are several **show** commands that display RA Guard information. They are documented in the *Network OS Command Reference*, and listed here with descriptions.

TABLE 22 RA Guard Show commands in the Network OS Command Reference

Command	Description
show ipv6 raguard	Displays RA Guard status on a specified interface or all interfaces on the device.
show running-config interface	Displays configuration information for an interface type or for a specific interface. RA Guard configuration is also displayed.