

Extreme Network OS Security Configuration Guide, 7.4.0

Supporting Network OS 7.4.0

Legal Notice

Extreme Networks, Inc. reserves the right to make changes in specifications and other information contained in this document and its website without prior notice. The reader should in all cases consult representatives of Extreme Networks to determine whether any such changes have been made.

The hardware, firmware, software or any specifications described or referred to in this document are subject to change without notice.

Trademarks

Extreme Networks and the Extreme Networks logo are trademarks or registered trademarks of Extreme Networks, Inc. in the United States and/or other countries.

All other names (including any product names) mentioned in this document are the property of their respective owners and may be trademarks or registered trademarks of their respective companies/owners.

For additional information on Extreme Networks trademarks, please see: www.extremenetworks.com/company/legal/trademarks

Open Source Declarations

Some software files have been licensed under certain open source or third-party licenses. End-user license agreements and open source declarations can be found at: www.extremenetworks.com/support/policies/software-licensing

Contents

Preface	7
Conventions.....	7
Notes, cautions, and warnings.....	7
Text formatting conventions.....	7
Command syntax conventions.....	8
Documentation and Training.....	8
Training.....	8
Getting Help.....	8
Subscribing to Service Notifications.....	9
Providing Feedback to Us.....	9
About this document	11
Supported hardware and software.....	11
Using the Network OS CLI	11
What's new in this document.....	12
User Accounts and Passwords	13
User account overview.....	13
Default accounts and roles.....	13
Account guidelines and limitations.....	13
Basic account management.....	14
Creating an admin-role account.....	14
Creating a user-role account.....	14
Modifying an account.....	14
Disabling an account.....	15
Unlocking an account.....	15
Deleting an account.....	16
User-defined roles.....	16
User-defined-role overview.....	16
Role and rule limits.....	16
Creating or modifying a role.....	17
Deleting a role.....	17
Creating a VCS Fabric security administrator role and account.....	17
Command-access rules.....	18
Rules for configuration commands.....	18
Rules for operational commands.....	18
Rules for interface commands.....	19
Configuring a placeholder rule.....	20
Rule-processing order.....	20
Adding a rule.....	20
Changing a rule.....	21
Deleting a rule.....	21
Advanced account management.....	21
Creating a non-default account.....	22
Creating an account with clock-restricted access.....	22
Password policies.....	22
Password policies overview.....	22

Configuring password policies.....	24
Password interaction with remote AAA servers.....	26
Security-event logs.....	26
User accounts and passwords show commands	26
Configuring Remote Server Authentication.....	27
Remote server authentication overview.....	27
Login authentication mode.....	27
Conditions for conformance.....	28
Configuring remote server authentication.....	28
Setting and verifying the login authentication mode.....	29
Resetting the login authentication mode.....	29
Changing the login authentication mode.....	29
Lightweight Directory Access Protocol.....	31
Understanding and configuring LDAP.....	31
User authentication.....	31
Server authentication.....	32
Server authorization.....	32
FIPS compliance.....	32
Configuring LDAP.....	33
Importing an LDAP CA certificate.....	33
Deleting LDAP CA certificates.....	33
Viewing the LDAP CA certificate.....	33
Configuring an Active Directory server on the client side.....	34
Adding an LDAP server to the client server list.....	34
Changing LDAP server parameters.....	35
Removing an LDAP server.....	35
Configuring Active Directory groups on the client side.....	35
Mapping an Active Directory group to a device role.....	36
Removing the mapping of an Active Directory to a device role.....	36
Configuring the client to use LDAP/AD for login authentication.....	36
Configuring an Active Directory server on the server side.....	36
Creating a user account on an LDAP/AD server.....	36
Verifying the user account on a device.....	37
Configuring LDAP users on a Windows AD server.....	37
RADIUS Server Authentication.....	39
RADIUS security.....	39
RADIUS Authentication.....	39
RADIUS Authorization.....	39
Account password changes.....	39
RADIUS authentication through management interfaces.....	39
Configuring server-side RADIUS support.....	40
Configuring RADIUS Server on a device.....	49
RADIUS two factor authentication support.....	52
TACACS+ Server Authentication.....	55
Understanding and configuring TACACS+	55
TACACS+ authentication, authorization, and accounting.....	55
Supported TACACS+ packages and protocols.....	55
TACACS+ configuration components.....	55

Client configuration for TACACS+ support.....	55
Client configuration for TACACS+ authorization.....	58
Client configuration for TACACS+ accounting.....	59
Configuring TACACS+ on the server side	62
Configuring TACACS+ for a mixed-vendor environment.....	65
HTTPS Certificates.....	67
HTTPS certificate overview.....	67
Configuring HTTPS certificates.....	67
Disabling HTTPS certificates.....	69
Enabling HTTPS service.....	70
Disabling HTTPS service.....	71
Importing TLS certificate and keys without trust point.....	71
ACLs.....	75
ACL overview.....	75
ACL application-targets.....	75
Interface ACLs and rACLs.....	76
ACLs applied to interfaces.....	77
ACL and rule limits.....	77
Layer 2 (MAC) ACLs.....	78
MAC ACL configuration guidelines.....	78
Creating a standard MAC ACL.....	79
Creating an extended MAC ACL.....	80
Applying Layer 2 ACLs to interfaces	80
Modifying MAC ACL rules.....	82
Reordering the sequence numbers in a MAC ACL.....	82
Creating MAC ACL rules enabled for counter statistics.....	83
ACL logs.....	83
Layer 3 (IPv4 and IPv6) ACLs.....	84
Implementation flow for rACLs and interface ACLs.....	84
Layer 3 ACL configuration guidelines.....	85
Creating a standard IPv4 ACL.....	87
Creating a standard IPv6 ACL.....	88
Creating an extended IPv4 ACL.....	88
Creating an extended IPv6 ACL.....	89
Applying Layer 3 ACLs to interfaces.....	89
Applying Layer 3 rACLs to RBridges.....	92
Modifying Layer 3 ACL rules.....	93
Reordering the sequence numbers in a Layer 3 ACL.....	93
ACL counter statistics (Layer 3).....	94
ACL logs.....	95
ACL Show and Clear commands.....	96
PBR - Policy-Based Routing.....	97
Policy-Based Routing.....	97
Notes:	98
Policy-Based Routing behavior.....	98
Policy-Based Routing with differing next hops.....	99
Policy-Based Routing uses of NULL0.....	100
Policy-Based Routing and NULL0 with match statements.....	100
Policy-Based Routing and NULL0 as route map default action.....	101

802.1x Port Authentication.....	103
802.1x protocol overview.....	103
Configuring 802.1x authentication.....	103
Understanding 802.1x configuration guidelines and restrictions.....	103
Configuring authentication	103
Configuring interface-specific administrative features for 802.1x.....	104
MAC authentication	108
MAC authentication bypass	108
Dynamic VLAN assignment in MAC authentication and MAB.....	109
Configuration notes for MAC authentication and MAB	110
Configuring MAC authentication bypass.....	110
Configuring MAC authentication.....	112
Port MAC Security.....	113
Port MAC security overview.....	113
Default port MAC security configuration options.....	113
Port MAC security commands.....	113
Port MAC security troubleshooting commands.....	114
Port MAC security guidelines and restrictions.....	114
Configuring port MAC security.....	115
Configuring port MAC security on an access port.....	115
Configuring port MAC security on a trunk port.....	115
Configuring port MAC security MAC address limits.....	115
Configuring port MAC security shutdown time.....	116
Configuring OUI-based port MAC security.....	116
Configuring port MAC security with sticky MAC addresses.....	117
SSH - Secure Shell.....	119
Configuring SSH encryption protocol	119
Configuring SSH ciphers.....	119
Configuring non-CBC SSH cipher.....	120
Removing an SSH cipher.....	121
Configuring SSH key-exchange.....	121
Removing an SSH key-exchange.....	122
Configuring SSH MAC.....	122
Removing an SSH MAC.....	123
Managing SSH public keys.....	123
Importing an SSH public key.....	124
Deleting an SSH public key.....	124
Configuring self-signed certificates for VXLAN gateways.....	124
Removing self-signed certificates for VXLAN gateways.....	124
Configuring the maximum number of SSH sessions.....	125
Router Advertisement (RA) Guard.....	127
RA Guard overview.....	127
RA Guard configuration guidelines	127
Enabling and disabling RA Guard	128
RA Guard Show commands.....	128

Preface

- Conventions..... 7
- Documentation and Training..... 8
- Getting Help..... 8
- Providing Feedback to Us..... 9

This section discusses the conventions used in this guide, ways to provide feedback, additional help, and other Extreme Networks® publications.

Conventions

This section discusses the conventions used in this guide.

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.

Text formatting conventions

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

Format	Description
bold text	Identifies command names. Identifies keywords and operands. Identifies the names of GUI elements.
<i>italic text</i>	Identifies text to enter in the GUI. Identifies emphasis. Identifies variables. Identifies document titles.

Format	Description
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.
<i>italic text</i>	Identifies a variable.
[]	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.
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, <i>member[member...]</i> .
\	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.

Documentation and Training

To find Extreme Networks product guides, visit our documentation pages at:

Current Product Documentation	www.extremenetworks.com/documentation/
Archived Documentation (for earlier versions and legacy products)	www.extremenetworks.com/support/documentation-archives/
Release Notes	www.extremenetworks.com/support/release-notes
Hardware/Software Compatibility Matrices	https://www.extremenetworks.com/support/compatibility-matrices/
White papers, data sheets, case studies, and other product resources	https://www.extremenetworks.com/resources/

Training

Extreme Networks offers product training courses, both online and in person, as well as specialized certifications. For more information, visit www.extremenetworks.com/education/.

Getting Help

If you require assistance, contact Extreme Networks using one of the following methods:

- Extreme Portal** Search the GTAC (Global Technical Assistance Center) knowledge base, manage support cases and service contracts, download software, and obtain product licensing, training, and certifications.
- The Hub** A forum for Extreme Networks customers to connect with one another, answer questions, and share ideas and feedback. This community is monitored by Extreme Networks employees, but is not intended to replace specific guidance from GTAC.
- Call GTAC** For immediate support: 1-800-998-2408 (toll-free in U.S. and Canada) or +1 408-579-2826. For the support phone number in your country, visit: www.extremenetworks.com/support/contact

Before contacting Extreme Networks for technical support, have the following information ready:

- Your Extreme Networks service contract number and/or serial numbers for all involved Extreme Networks products
- A description of the failure
- A description of any action(s) already taken to resolve the problem
- A description of your network environment (such as layout, cable type, other relevant environmental information)
- Network load at the time of trouble (if known)
- The device history (for example, if you have returned the device before, or if this is a recurring problem)
- Any related RMA (Return Material Authorization) numbers

Subscribing to Service Notifications

You can subscribe to email notifications for product and software release announcements, Vulnerability Notices, and Service Notifications.

1. Go to www.extremenetworks.com/support/service-notification-form.
2. Complete the form with your information (all fields are required).
3. Select the products for which you would like to receive notifications.

NOTE

You can modify your product selections or unsubscribe at any time.

4. Click **Submit**.

Providing Feedback to Us

Quality is our first concern at Extreme Networks, and we have made every effort to ensure the accuracy and completeness of this document. We are always striving to improve our documentation and help you work better, so we want to hear from you! We welcome all feedback but especially want to know about:

- Content errors or confusing or conflicting information.
- Ideas for improvements to our documentation so you can find the information you need faster.
- Broken links or usability issues.

If you would like to provide feedback to the Extreme Networks Information Development team, you can do so in two ways:

- Use our short online feedback form at <https://www.extremenetworks.com/documentation-feedback/>.
- Email us at documentation@extremenetworks.com.

Please 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..... 11
- Using the Network OS CLI 11
- What's new in this document..... 12

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 Extreme Networks, 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:

- ExtremeSwitching VDX 6740-48
- ExtremeSwitching VDX 6740T
 - ExtremeSwitching VDX 6740T-64
 - ExtremeSwitching VDX 6740T-1G
- ExtremeSwitching VDX 6940-144S
- ExtremeSwitching VDX 6940-36Q
- ExtremeSwitching VDX 8770
 - ExtremeSwitching VDX 8770-4
 - ExtremeSwitching 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 Extreme Network OS command line interface (CLI), refer to the *Extreme Network OS Command Reference*.

What's new in this document

The following table describes new information added to this document for the Network OS 7.4.0 software release.

NOTE

Fibre Channel (FC) is no longer supported; commands related to FC and "FCoE" (Fibre Channel over Ethernet) have been either removed or modified. However, instances of "FC" and "FCoE" and related services may still appear in CLI "show" outputs and elsewhere.

TABLE 1 Summary of enhancements in Network OS 7.4.0 release

Feature	Description	Described in
AAA command authorization (TACACS+)	AAA command authorization is supported for TACACS+	Client configuration for TACACS+ authorization on page 58

For complete information, refer to the *Network OS 7.4.0 Release Notes*.

User Accounts and Passwords

• User account overview.....	13
• Basic account management.....	14
• User-defined roles.....	16
• Command-access rules.....	18
• Advanced account management.....	21
• Password policies.....	22
• Security-event logs.....	26
• User accounts and passwords show commands	26

User account overview

A user account specifies that user's level of access to the device CLI.

The software uses role-based access control (RBAC) as the authorization mechanism. A *role* is a container for rules, which specify which commands can be executed and with which permissions. When you create a user account you need to specify a role for that account. In general, *user* (as opposed to *user-level*) refers to any account—to which any role can be assigned—user, admin, or a non-default role.

Default accounts and roles

The software ships with two default accounts—admin and user—and two corresponding default roles:

- **admin**—Accounts with admin permissions can execute all commands supported on the device. (For the initial admin login, refer to the relevant *Hardware Installation Guide*.)
- **user**—Accounts with user-level permissions can execute all **show** commands supported on the device. User-level accounts can also execute the following operational commands: **exit**, **ping**, **ssh**, **telnet**, **timestamp**, **rasman**, and **traceroute**.

NOTE

For details on non-default roles (also known as *user-defined roles*), refer to [User-defined roles](#) on page 16.

Account guidelines and limitations

Be aware of the following guidelines and limitations:

- Extreme recommends that every user access the CLI through a unique account: After logging in as admin, create a unique account for yourself, specifying **role admin**.
- You cannot modify rules for the admin or the user default accounts.
- You cannot modify rules for the admin or the user default roles.
- By default, all account information is stored in the device-local user database.
- By default, user authentication and tracking of logins to the device is local.
- The maximum number of accounts—including the two default accounts—is 64. For more than 64 users, you can implement an authentication, authorization, and accounting (AAA) service. For details, refer to the External Server Authentication section.
- The maximum number of roles—including the two default roles—is 64. If needed, refer to [Role and rule limits](#) on page 16.
- Role configuration is applied to all VCS nodes.

Basic account management

These topics enable you to create and manage basic admin and user accounts.

Creating an admin-role account

An admin-role account can execute all supported CLI commands.

The required parameters for creating an account are **name**, **role**, and **password**. In this example, the optional **desc** parameter is also utilized.

1. In privileged EXEC mode, enter the **configure terminal** command.

```
device# configure terminal
```

2. Enter the **username** command, with the specified parameters.

```
device(config)# username jsmith role admin password Tijdlspw desc "Has access to all commands"
```

Creating a user-role account

A user-role account can execute **show** and other basic CLI commands.

1. In privileged EXEC mode, enter the **configure terminal** command.

```
device# configure terminal
```

2. Enter the **username** command, with the specified parameters.

```
device(config)# username jdoe role user password iKt1Sas*p
```

Modifying an account

Use this topic to modify a user account.

The only required parameter for modifying an account is **username** *username*. In this example, the role is changed to admin.

1. In privileged EXEC mode, enter the **configure terminal** command.

```
device# configure terminal
```

2. Enter the **username** command, with the needed parameters.

```
device(config)# username jdoe role admin
```

Disabling an account

Use this topic to disable a user account.

NOTE

If you disable an account, all active sessions for that user are immediately terminated.

1. In privileged EXEC mode, enter the **configure terminal** command.

```
device# configure terminal
```

2. Enter the **username** command, with the **enable false** parameters.

```
device(config)# username testUser enable false
```

Unlocking an account

Use this topic to unlock 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 24 for more information.

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

NOTE

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

1. In privileged EXEC mode, enter the **show users** command to display currently active sessions and locked out users.

```
device# show users
rbridge-id
all
**USER SESSIONS**
ID Username Role Host IP Method Time Logged In TTY
2 jsmith user 192.0.2.0 cli 2016-04-30 01:59:35 pts/2
1 jdoe admin 192.0.2.1 cli 2016-05-30 01:57:41 tty80

**LOCKED USERS**
RBridge ID Username
1 testUser
```

2. For each account that you want to unlock, enter the **unlock username** command.

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

3. Enter the **show users** command to verify that the account is unlocked.

```
device# show users
rbridge-id
all
**USER SESSIONS**
ID Username Role Host Ip Method Time Logged In TTY
2 jsmith user 192.0.2.0 cli 2016-04-30 01:59:35 pts/2
1 jdoe admin 192.0.2.1 cli 2016-05-30 01:57:41 tty80

**LOCKED USERS**
RBridge ID Username
no locked users
```

Deleting an account

Use this topic to delete a user account.

1. In privileged EXEC mode, enter the **configure terminal** command.

```
device# configure terminal
```

2. Enter the **no username** command.

```
device(config)# no username testUser
```

When an account is deleted, all active login sessions for that user are terminated

User-defined roles

In addition to the default roles—admin and user—the software supports the creation of user-defined roles.

User-defined-role overview

User-defined roles enable you to fine-tune CLI access.

A user-defined role starts from a basic set of privileges which are then refined by adding rules. You 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:

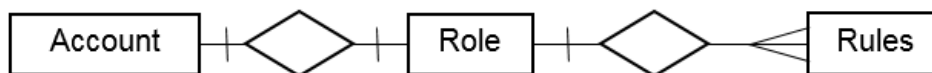
- The **role** command defines new roles and deletes user-defined roles.
- The **rule** command allows you to specify access rules for specific operations and assign these rules to a given role.
- The **username** command associates a given user-defined role with a specific user account.

Role and rule limits

At any given time, an account is associated with one role. A role is associated with one or more rules. A rule is associated with only one role.

This relationship among accounts, roles, and rules is illustrated by the following entity-relationship diagram:

FIGURE 1 Accounts, roles, and rules



The number of supported accounts, roles, and rules is as follows:

- The maximum number of accounts is 64, including the default admin and user accounts. For more than 64 users, you can implement an authentication, authorization, and accounting (AAA) service.
- The maximum number of roles is 64, including the default admin and user roles.
- The maximum number of rules is 512, which you can allocate among your roles as you see fit.

Creating or modifying a role

Use this topic to create a role or to modify its Description.

1. In privileged EXEC mode, enter the **configure terminal** command.

```
device# configure terminal
```

2. Enter the **role** command, specifying the role name and (optionally) a description.

```
device(config)# role name NetworkAdmin desc "Manages security CLIs"
```

Deleting a role

Use this topic to delete a role.

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

```
device# configure terminal
Entering configuration mode terminal
```

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

```
device(config)# no role name NetworkAdmin
```

Creating a VCS Fabric security administrator role and account

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

1. Create a role for an Extreme VCS Fabric security administrator.

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

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

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

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

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

Command-access rules

Command authorization is defined in terms of rules that you associate with a user-defined 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.

The following rule parameters are mandatory:

- **index**—a unique index number
- **role**—the unique role with which you are associating the rule
- **command**—the command to which the rule applies

The following rule parameters are optional:

- **operation**—specifies the type of operation permitted (**read-only** or **read-write**). The default is **read-write**.
- **action**—specifies whether the user is accepted or rejected while attempting to execute the specified command. The default value is **accept**.

The following example creates and assigns four rules to a role named "NetworkAdmin".

```
device(config)# rule 70 action accept operation read-write role NetworkAdmin command configure
device(config)# rule 71 action accept operation read-write role NetworkAdmin command copy running-config
device(config)# rule 72 action accept operation read-write role NetworkAdmin command interface management
device(config)# rule 73 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 `?` after the **command** keyword.

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.

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.

Rules for interface commands

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

The following rules govern interface commands:

- If a role has a rule with a **read-write** operation and the **accept** action for only a particular instance of the interface, users 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, users 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, users 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.

```
device(config)# rule 60 action accept operation read-write role NetworkAdmin command interface
tengigabitethernet 1/0/4
device(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, users 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, NetworkAdmin users cannot perform **clear** and **show** operations related to all **tengigabitethernet** instances.

```
device(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** instances, users associated with this role cannot perform **clear** and **show** operations related to those instances. To perform **clear** and **show** operations related to **interface tengigabitethernet** instances, the 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, users associated with the NetworkAdmin role cannot perform some of the **clear** and **show** operations related to all **tengigabitethernet** instances.

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

- 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, users associated with the CfgAdmin role can access and execute the **dot1x** command in **tengigabitethernet** instances.

```
device(config)# rule 16 action accept operation read-write role cfgadmin command interface
tengigabitethernet
device(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, users can perform the **no vlan** and **no spanning-tree** operations on the other **interface** instances for which users have 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.

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

```
device# configure terminal
```

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

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

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

```
device(config)# rule 75 role NetworkAdmin command firmware
```

Rule-processing order

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.

```
device(config)# rule 9 operation read-only action accept role NetworkAdmin command aaa
device(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.

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

```
device# configure terminal
```

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

```
device(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.

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

4. "SecAdminUser" users can create or modify user accounts.

```
device# 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

device(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.

```
device# configure terminal
```

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

```
device(config)# rule 155 role SecAdminUser command role
```

After changing rule 155, "SecAdminUser" users can execute the **role** command, but not the **username** command.

Deleting a rule

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

```
device# configure terminal
Entering configuration mode terminal
```

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

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

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

Advanced account management

These topics enable you to create non-default accounts and to configure advanced settings.

Creating a non-default account

The permissions for a non-default account are determined by the role assigned to it.

The required parameters for creating an account are **name**, **role**, and **password**. In this example, the optional **desc** parameter is also utilized.

1. In privileged EXEC mode, enter the **configure terminal** command.

```
device# configure terminal
```

2. Enter the **username** command, with the name, role, initial password, and optional parameters.

```
device(config)# username mlopez role NetworkAdmin password xL*84qt desc "Has access to all network admin commands."
```

Creating an account with clock-restricted access

When defining or editing an account, you can specify permitted access hours.

By default, users can log in 24 hours a day. The **access-time** parameter enables you to limit access to defined hours, as per the system time defined for the operating system. For the current system time, enter **show clock**.

1. In privileged EXEC mode, enter the **configure terminal** command.

```
device# configure terminal
```

2. Enter the **username** command, with the **access-time** parameter.

```
device(config)# username aming role user password Tijd1spw access-time 0800 to 1800
```

Password policies

Password policies define and enforce a set of rules that make passwords more secure by subjecting all new passwords to global restrictions.

Password policies overview

You can configure password strength policy, password encryption policy, and account lockout policy.

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.

Password configuration is applied to all nodes in the fabric.

NOTE

For recovering the root password, refer to the *Extreme Network OS Troubleshooting Guide*.

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. 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 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 enabled.

The following rules apply to password encryption:

- When you enable password encryption, all existing clear-text passwords will be encrypted, and any passwords 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.

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

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

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

device(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 device. 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.

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

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

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

device(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 device only. The user account is locked only on the device where the lockout occurred; the same user can still try to log in on another device 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

Extreme advises that you periodically examine the Security Audit logs to determine if such attacks are attempted. Refer to [Security-event logs](#) on page 26.

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.

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

When a user account is locked, it can be unlocked using the procedure described in [Unlocking an 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.

```
device# configure terminal
Entering configuration mode terminal
device(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.

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

Displaying password attributes

To display configured password attributes, change to privileged EXEC mode and enter **show running-config password-attributes**. Refer to the **password-attributes** command in the command reference for details on modifying password attributes.

```
device# 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
```

Password interaction with remote AAA servers

The password policies apply to local device 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.

Security-event logs

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 Extreme VCS Fabric mode, for fabric-wide events, the audit is generated on all devices of the fabric. Refer to the *Extreme Network OS Message Reference* for information on how to configure, monitor, and analyze security audit logging.

User accounts and passwords show commands

There are **show** commands that display user account and password information, listed here with descriptions.

TABLE 3 User account and password show commands in the *Command Reference*

Command	Description
show running-config password-attributes	Displays global password attributes.
show running-config role	Displays name and description of the configured roles.
show running-config rule	Displays configured access rules.
show running-config username	Displays the user accounts on the device.
show users	Displays the users logged in to the system and locked user accounts.

Configuring Remote Server Authentication

- [Remote server authentication overview.....](#) 27
- [Configuring remote server authentication.....](#) 28

Remote server authentication overview

The software supports various protocols to provide external Authentication, Authorization, and Accounting (AAA) services for 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 device acts as a network access server client. The device 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 device.

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 device 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.

Extreme 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 device. Each device maintains its own server configuration.

Login authentication mode

The authentication mode is defined as the order in which AAA services are used on the device 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 device-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 device-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 device 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-fallback** command. In the following example, the local device database will be used if the RADIUS server is unavailable.

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

Conditions for conformance

Consider the following conditions for remote server authentication:

- 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+ in a simple manner.

For detailed configuration information on remote server authentication, refer to the following topics:

- [RADIUS security](#) on page 39
- [Understanding and configuring TACACS+](#) on page 55
- [Understanding and configuring LDAP](#) on page 31

Setting and verifying the login authentication mode

The following procedure configures TACACS+ as the primary source of authentication and the device-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.

```
device# configure terminal
Entering configuration mode terminal
```

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

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

Broadcast message from root (pts/0) Tue Apr 5 16:34:12 2016...
AAA Server Configuration Change: all accounts will be logged out
```

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

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

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

Resetting the login authentication mode

The following procedure resets the login configuration mode to the default value using the **no aaa authentication login** command.

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

```
device# 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).

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

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

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

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

You can reset the configuration to the default value using the **no aaa authentication login** command.

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.

```
device# configure terminal
Entering configuration mode terminal
```

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

```
device(config)# aaa authentication login radius local
Broadcast message from root (pts/0) Tue Apr 5 16:34:12 2016...
AAA Server Configuration Change: all accounts will be logged out
```

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

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

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

Lightweight Directory Access Protocol

- Understanding and configuring LDAP..... 31
- Configuring LDAP..... 33
- Importing an LDAP CA certificate..... 33
- Deleting LDAP CA certificates..... 33
- Viewing the LDAP CA certificate..... 33
- Configuring an Active Directory server on the client side..... 34
- Configuring Active Directory groups on the client side..... 35
- Configuring an Active Directory server on the server side..... 36

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 that supports a number of standardized protocols such as LDAP, Kerberos authentication, and Domain Name Server (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 part of directory information, so it can be used as a centralized database for authenticating third-party resources.

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

User authentication

A 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, the device supports server authentication during the TLS handshake. Only the user principal name from the AD server is supported for LDAP authentication on the device. The common name (CN) 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 device configured to perform LDAP-based authentication supports 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 device.

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

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

Server authentication

As a part of user authentication using LDAP, the 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 must be added as the host parameter, instead of the IP address. An 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 must 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 must be imported, or CA certificates must 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 must 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 must be assigned a nonprimary group, and that group name must 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 the AD server (or default role) is used for RBAC.

If multiple nonprimary groups are associated to the AD user, only one of the groups must 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 the 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 of 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 must be installed on the device, and the FIPS-compliant TLS ciphers for LDAP must be used.

Configuring LDAP

Configuring support for LDAP requires configuring both the client and the server. The following major tasks are sorted by client-side and server-side activities:

Client-side tasks:

- [Configuring an Active Directory server on the client side](#) on page 34
- [Configuring Active Directory groups on the client side](#) on page 35

Server-side tasks:

- [Creating a user account on an LDAP/AD server](#) on page 36
- [Verifying the user account on a device](#) on page 37
- [Configuring LDAP users on a Windows AD server](#) on page 37

Importing an LDAP CA certificate

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

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

```
device# configure terminal
Entering configuration mode terminal
```

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

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

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

```
device# show cert-util ldapcert
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.

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

Viewing the LDAP CA certificate

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

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

```
device# show cert-util ldapcert
```

Logical chassis fabric mode

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

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

Configuring an Active Directory server on the client side

Each device client must be individually configured to use Active Directory servers. You can configure a maximum of five Active Directory servers on a device for AAA service.

You use the **ldap-server** command to specify the host server, authentication protocols, and other parameters.

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

TABLE 4 Active Directory parameters

Parameter	Description
host	IPv4 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.
use-vrf	Specifies a VRF through which to communicate with the Active Directory server.

Adding an LDAP server to the client server list

The following procedure configures an LDAP server on an LDAP client device.

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

```
device# 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 submenu where you can modify the server default settings.

```
device(config)# ldap-server host 10.24.65.6
device(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 table in [Configuring an Active Directory server on the client side](#) on page 34).

```
device(config-ldap-server 10.24.65.6)# basedn security.brocade.com
device(config-ldap-server 10.24.65.6)# timeout 8
device(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.

```
device(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 global configuration mode.

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

Changing LDAP server parameters

Changing the LDAP server parameters 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, and then enter the new values as required.

Refer to [Adding an LDAP server to the client server list](#) on page 34.

```
device# configure terminal
Entering configuration mode terminal
device(config)# ldap-server host 10.24.65.6
device(config-host-10.24.65.6/mgmt-vrf)# basedn security.brocade.com
```

Removing an LDAP server

The following procedure deletes an LDAP server entry from the device LDAP server list.

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

```
device# configure terminal
Entering configuration mode terminal
```

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

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

Configuring Active Directory groups on the client side

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

A user on an AD server must be assigned a nonprimary group, and that group name must 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 a role from any nonprimary group.

Mapping an Active Directory group to a device role

In the following example, a 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.

```
device# 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 device roles.

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

Removing the mapping of an Active Directory to a device role

The following example removes the mapping between the Extreme admin role and the Active Directory (AD) Administrator group. A 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.

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

```
device# configure terminal
Entering configuration mode terminal
```

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

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

Configuring the client to use LDAP/AD for login authentication

After you configure the device LDAP server list, you must set the authentication mode so that LDAP is used as the primary source of authentication.

Refer to [Login authentication mode](#) on page 27 for information on how to configure the login authentication mode.

Configuring an Active Directory server on the server 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

The following procedure configures 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 match with the user's Extremedevic role.

3. Optional: You can map the role to the Extreme device role with the **ldap-server maprole** command.
4. Associate the user with the group by adding the user to the group.
The user account configuration is complete.

Verifying the user account on a device

The following procedure verifies a user account on a device.

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

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

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

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

4. Log in to the device 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 device using an account with device-local only credentials. The login should fail with an access denied message.

Configuring LDAP users on a Windows AD server

The following procedure configures a user account on a Windows AD server.

1. Create a user in Windows.
 - a) Open **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 in Windows.
 - 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 device role or it must be mapped with the device role on the device. In this instance, Domain Users is the primary group and therefore should not be mapped with the device role.

RADIUS Server Authentication

- RADIUS security..... 39

RADIUS security

The remote authentication dial-in user service (RADIUS) protocol manages authentication, authorization, and accounting (AAA) services centrally.

You can use a Remote Authentication Dial In User Service (RADIUS) server to secure the following types of access to the 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, using roles pre-defined on the device and sent as attribute in Radius Response.

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

RADIUS Authentication

When RADIUS authentication is implemented, the device consults a RADIUS server to verify user names and passwords.

During the user authentication process, the device sends its IP address. When the device also has a Virtual IP address (in Extreme VCS Fabric mode), it still sends only the unique IP address of the node that you are logging in to, to the RADIUS server.

RADIUS Authorization

User authorization through the RADIUS protocol is not supported. The access control of RADIUS users is enforced by the Extreme 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 be assigned the "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 (Ethernet 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 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 an Extreme Vendor-Specific Attribute (VSA). An Authentication-Accept response without the role assignment automatically grants the "user" role.

NOTE

RADIUS Server must be configured to support Vendor-Specific-Attribute (VSA) in addition to configuring RADIUS Server support on the device.

Configuring a RADIUS server with Linux

FreeRADIUS is an open source RADIUS server that runs on all versions of Linux (FreeBSD, NetBSD, and Solaris).

Perform the following steps to configure a RADIUS server with Linux.

1. Download the package from www.freeradius.org and follow the installation instructions at the FreeRADIUS website.
2. Refer to the RADIUS product documentation for information on configuring and starting up a RADIUS server.
3. Determine where vendor-specific dictionaries are located on the server.

```
user@Linux:$ locate dictionary.*
/usr/share/freeradius/dictionary.3com
/usr/share/freeradius/dictionary.3gpp
/usr/share/freeradius/dictionary.3gpp2
/usr/share/freeradius/dictionary.acc
/usr/share/freeradius/dictionary.acme
```

4. Change to the vendor-specific dictionaries directory.

```
user@Linux:$ cd /usr/share/freeradius/
user@Linux: /usr/share/freeradius$
```

5. Verify that the `dictionary.brocade` file exists in this directory.

```
user@Linux: /usr/share/freeradius$ ls dictionary.brocade
dictionary. brocade
```

When the `dictionary.brocade` file does not exist, proceed to Step 7.

6. Check that the contents of the `dictionary.brocade` file are correct. The following example shows the correct information.

```
user@Linux: /usr/share/freeradius$ more dictionary.brocade
# -*- text -*-
# Copyright (C) 2013 The FreeRADIUS Server project and contributors
#
VENDOR          Brocade          1588
BEGIN-VENDOR    Brocade

ATTRIBUTE       Brocade-Auth-Role 1      string

END-VENDOR      Brocade
```

When the `dictionary.brocade` file exists and holds the correct information, proceed to Step 10.

7. When the `dictionary.brocade` file does not exist or holds incorrect information, you need to create a `dictionary.brocade` file with the correct information.
 - a) Log in as the root user.
 - b) In the vendor-specific dictionaries directory, create a file named `dictionary.brocade` with the below content.

```
# -*- text -*-
# Copyright (C) 2013 The FreeRADIUS Server project and contributors
#
VENDOR          Brocade          1588
BEGIN-VENDOR    Brocade

ATTRIBUTE       Brocade-Auth-Role 1 string

END-VENDOR      Brocade
```

8. To import the `dictionary.brocade` file, add the following line to the dictionary file.

```
$INCLUDE dictionary.brocade
```

9. To ensure that the dictionary is loaded, restart the FreeRADIUS server.

```
user@Linux:/usr/share/freeradius$ sudo service freeradius restart
```

10. Configure an Extreme user account.

- a) Open the `/etc/raddb/users` file in a text editor (the location of the FreeRADIUS users configuration file depends on the Linux distribution).
- b) Add the user name and associated the permissions. You must log in as `rootadmin` using admin permissions specified with `Brocade-Auth-Role`. The following example shows how to configure the `rootadmin` user account with a password "passadmin", a Service-Type of `Framed-User`, and admin permissions.

```
rootadmin Cleartext-Password := "passadmin"
        Service-Type = Framed-User,
        Brocade-Auth-Role = "admin"
```

NOTE

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

11. To ensure that the changes take effect, restart the FreeRADIUS server.

```
user@Linux:/usr/share/freeradius$ sudo service freeradius restart
```

NOTE

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

Configuring a Windows NPS-based RADIUS server

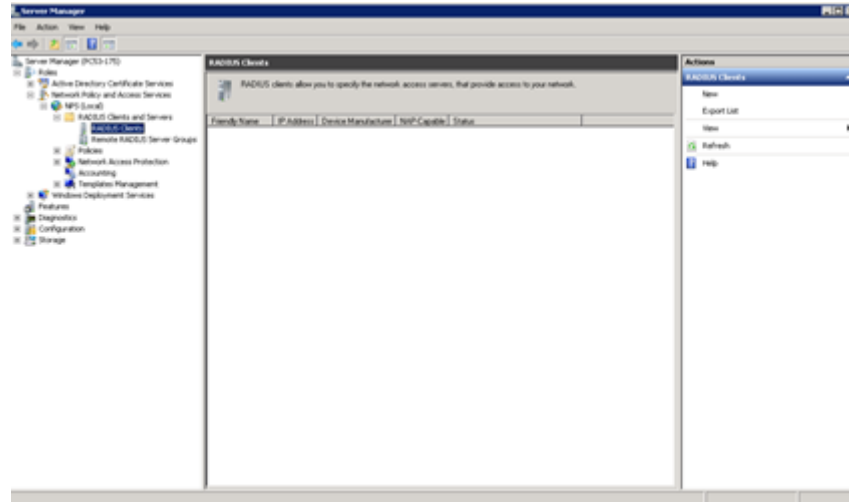
Step-by-step instructions for installing and configuring Network Policy Server (NPS) with Microsoft Windows server 2008 (or later; for example Windows 2012) can be obtained from www.microsoft.com or your Microsoft documentation. Confer with your system or network administrator prior to configuration for any special needs your network environment may have.

Use the following information to configure the Network Policy Server for a device.

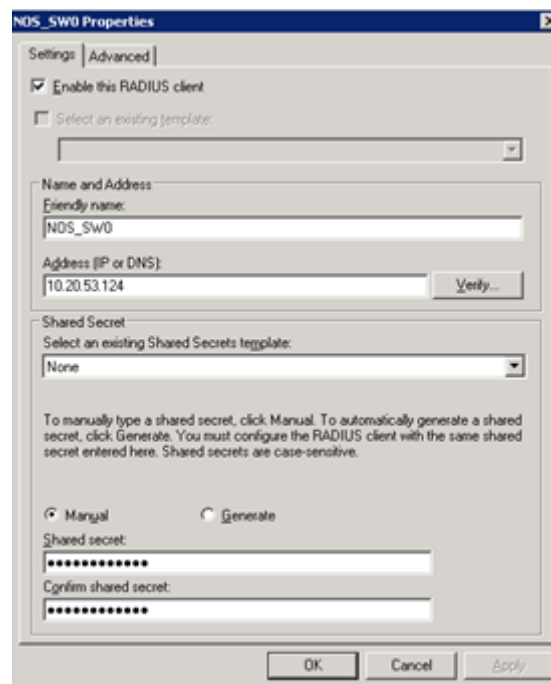
NOTE

This is not a complete presentation of steps.

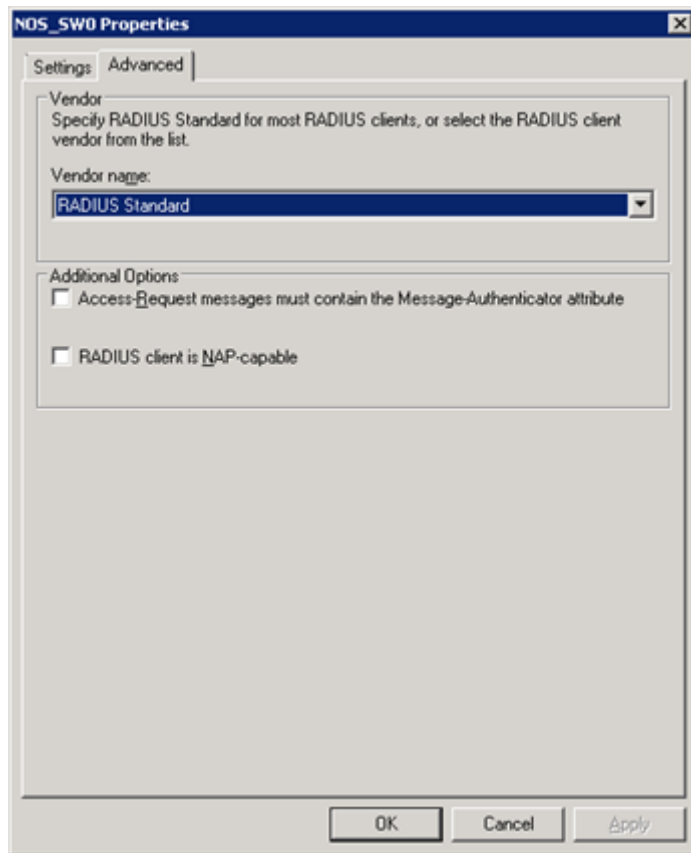
1. Configure a new RADIUS client.
 - a) Open **Server Manager**, expand **Roles**, expand **Network Policy and Access Services**, expand **NPS**, expand **RADIUS Clients and Servers** and click on **RADIUS Clients**. Then, on the **Actions** panel, click **New**.



- b) On the **Settings** tab, select **Enable this RADIUS client** and configure the **Friendly name**, **Address (IP or DNS)**, and **Shared Secret** for the RADIUS client.



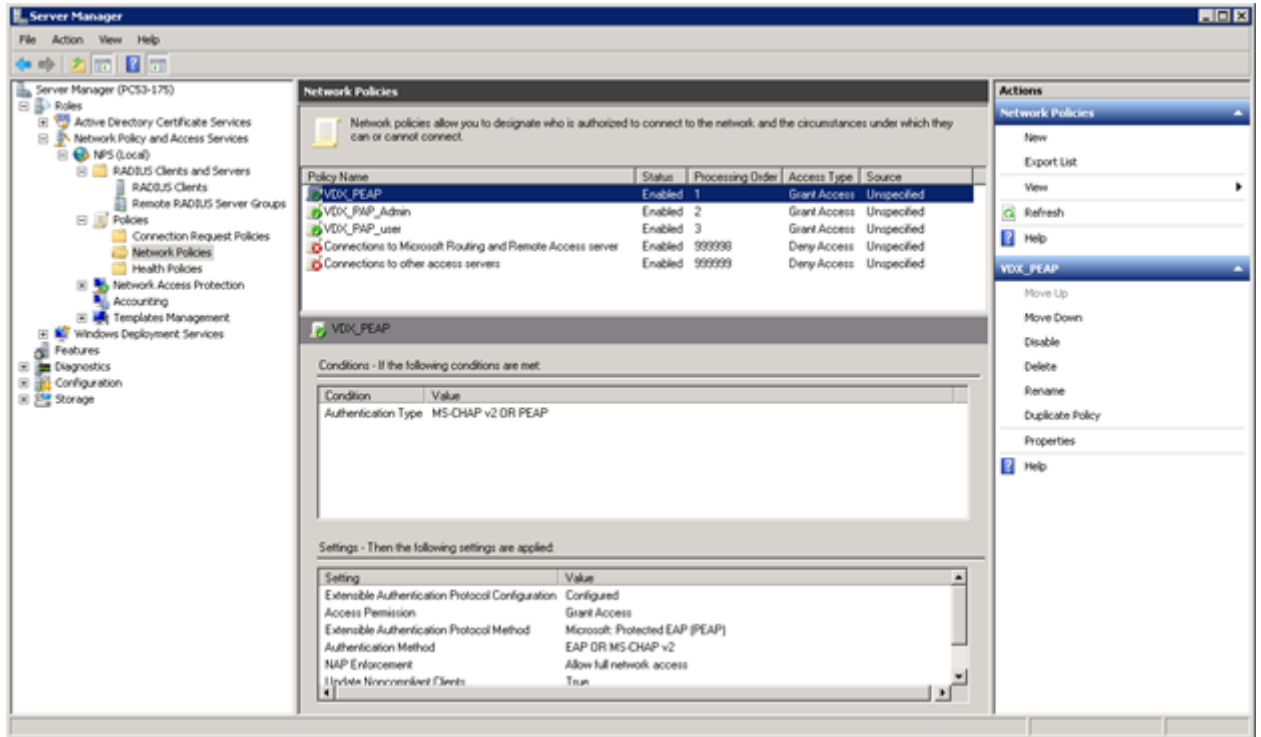
- c) On the **Advanced** tab, check that the **Vendor name** is set to **RADIUS Standard**. Leave **Additional Options** blank (unless you have configured your network policies to support these). Then, click **OK**.



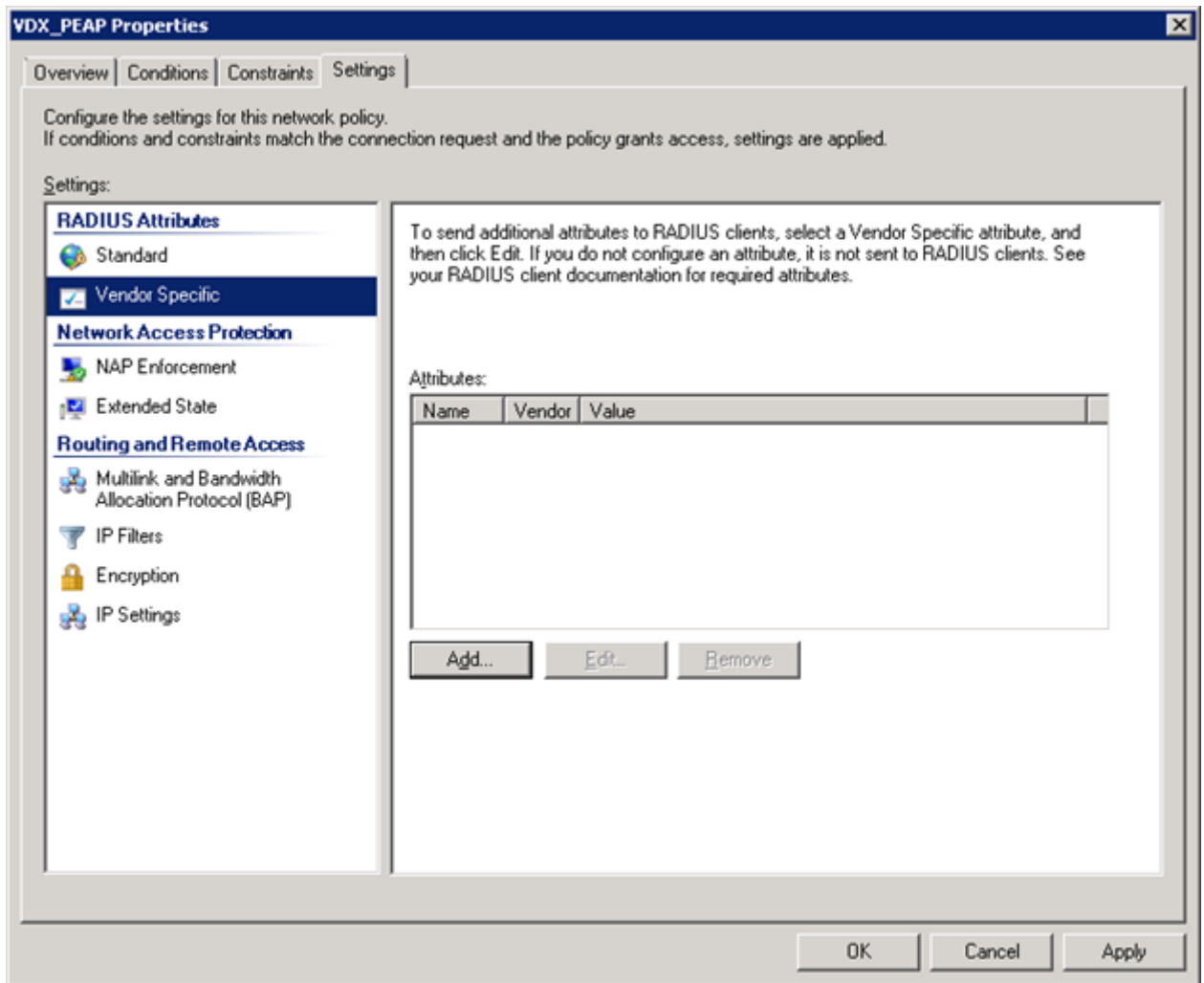
2. Configure Vendor Specific Attributes (VSA) in Network Policies.

The following steps describe how to add the VSA for a NOS device to your network policy; for further information about creating network policies, refer to www.microsoft.com or your Microsoft documentation.

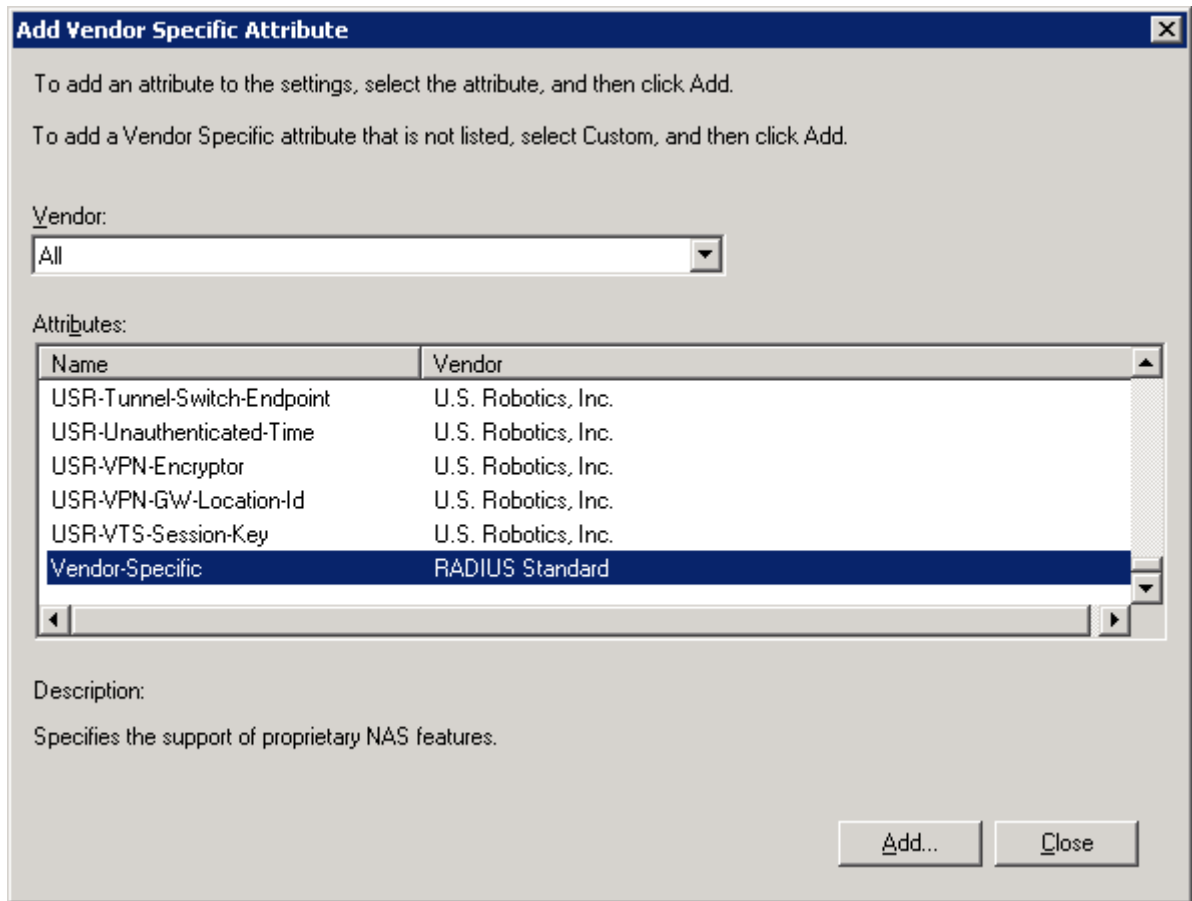
- a) In **Server Manager**, expand **Roles, Network Policy and Access Services, NPS, and Policies**. Then click on **Network Policies** and select the authorization policy that you created. On the **Actions** panel, click **Properties**.



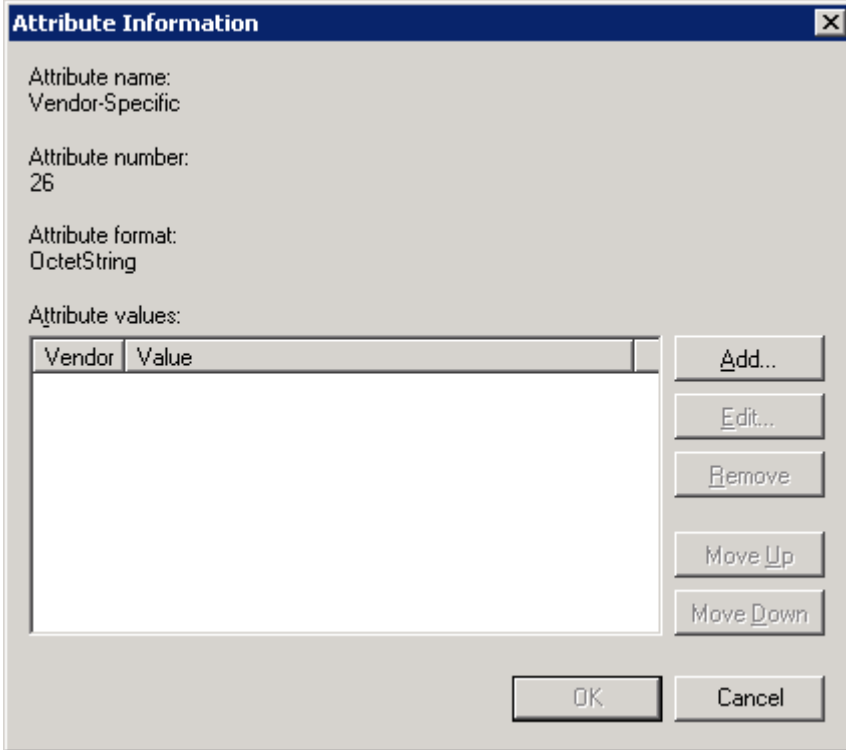
- b) On the **Settings** tab, select **Vendor Specific** and click **Add**.



- c) Scroll down through the **Attributes** and select **Vendor Specific**. Then click **Add**.



- d) On the **Attribute Information** screen, click **Add**.



Attribute Information

Attribute name:
Vendor-Specific

Attribute number:
26

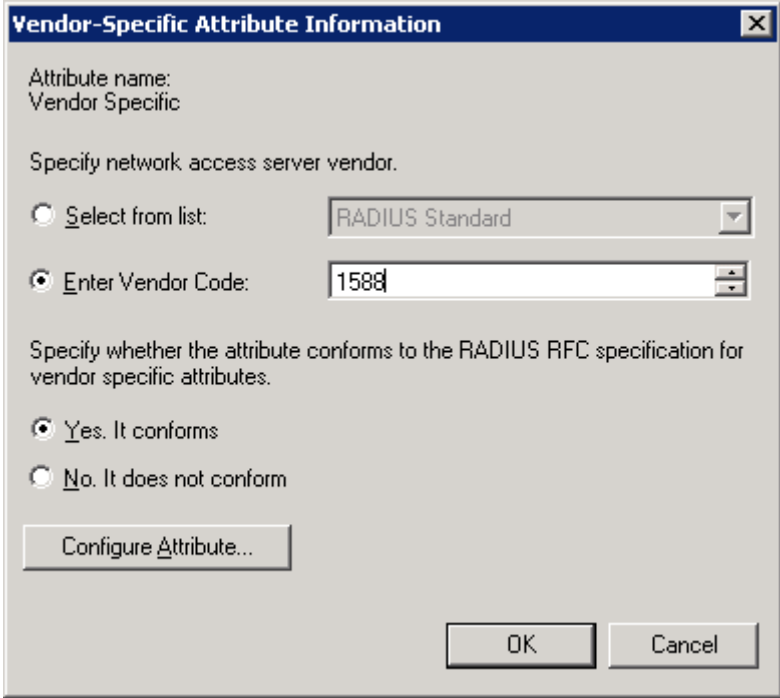
Attribute format:
OctetString

Attribute values:

Vendor	Value

Buttons: Add..., Edit..., Remove, Move Up, Move Down, OK, Cancel

- e) On the **Vendor-Specific Attribute Information** screen, configure the **Vendor Code** as 1588 and check **Yes: It conforms**. Then, click **Configure Attribute**.



Vendor-Specific Attribute Information

Attribute name:
Vendor Specific

Specify network access server vendor.

Select from list: RADIUS Standard

Enter Vendor Code: 1588

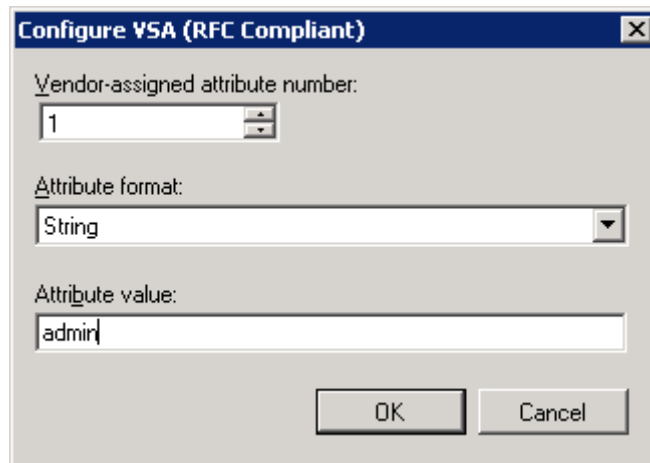
Specify whether the attribute conforms to the RADIUS RFC specification for vendor specific attributes.

Yes. It conforms

No. It does not conform

Buttons: Configure Attribute..., OK, Cancel

- f) On the **Configure VSA (RFC Compliant)** screen, set **Vendor-assigned attribute number** to 1, **Attribute format** to String, and **Attribute value** to admin (for admin user) or user (for default user).



- g) Click **OK** on each screen until you return to the **Add Vendor Specific Attribute** screen. Then click **Close** and **Close** again on the network policy properties screen.

Configuring a Windows IAS-based RADIUS 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 device.

NOTE

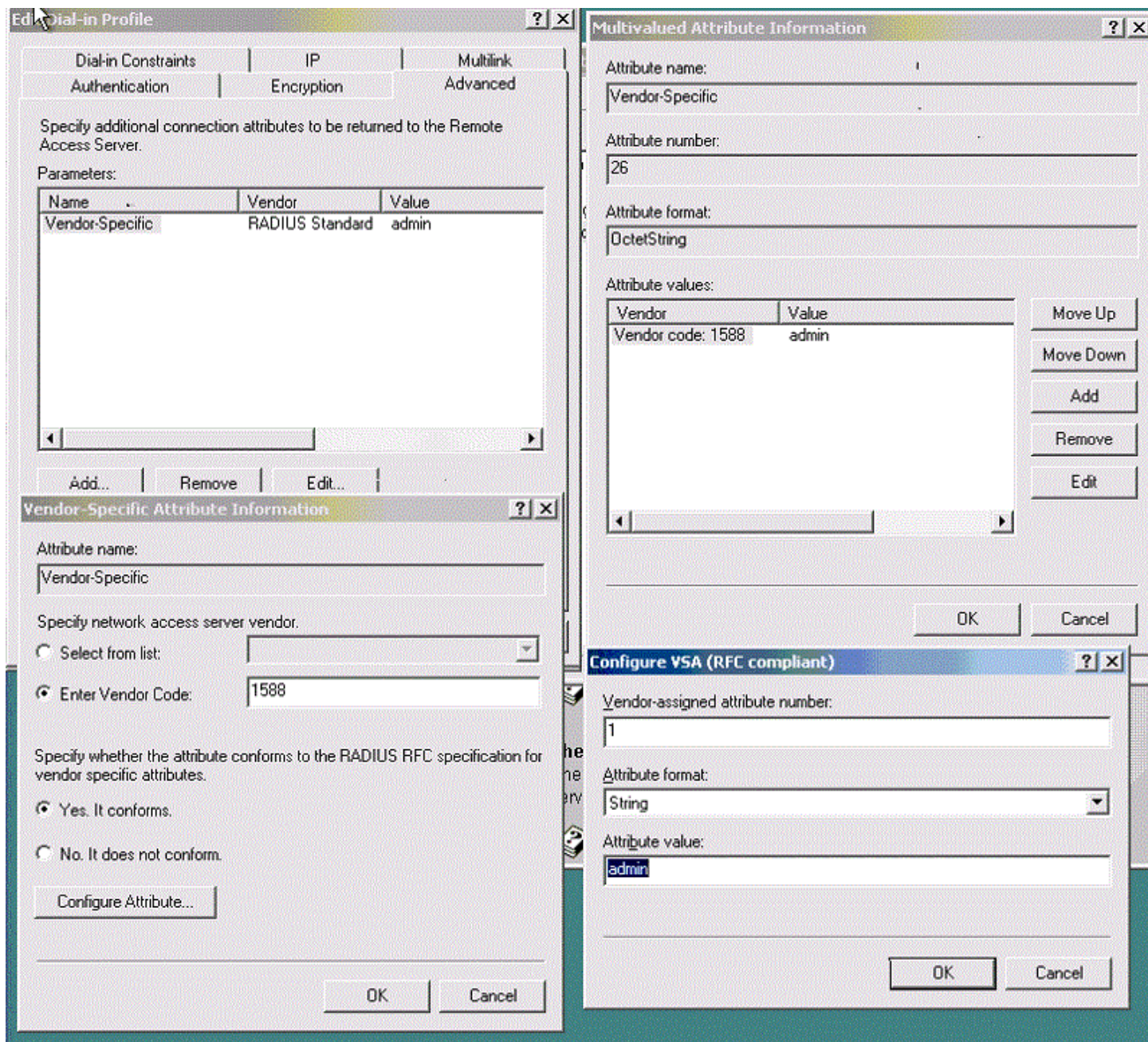
This is not a complete presentation of steps.

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 Extreme vendor code value.
 - 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 2 Windows server VSA configuration



Configuring RADIUS Server on a device

Each device client must be individually configured to use RADIUS servers.

You use the **radius-server host** command to specify the server IP address, authentication protocols, and other parameters.

You can configure a maximum of 5 RADIUS servers on a device for AAA service.

NOTE

RADIUS Server must be configured to support Vendor-Specific-Attribute (VSA) in addition to configuring RADIUS Server support on the device.

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

TABLE 5 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 255 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.
use-vrf	Specifies a VRF though which to communicate with the RADIUS server.

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](#) on page 50
- [Modifying the RADIUS server configuration](#) on page 51
- [Configuring the client to use RADIUS for login authentication](#) on page 52

Adding a RADIUS server

You can configure up to five RADIUS servers on a device.

Prior to configuring a RADIUS server by specifying a domain or host name, you must configure the Domain Name System (DNS) server on the device by using the **ip dns** command. The host name cannot be resolved unless the DNS server is configured.

NOTE

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

Perform the following task to add a RADIUS server to a device.

1. From privileged EXEC mode, enter global configuration mode.

```
device# configure terminal
Entering configuration mode terminal
```

- When the default configuration values for communication with the RADIUS server are not acceptable, use the **radius-server host** command specifying the **use-vrf** parameter to enter RADIUS server host VRF configuration mode.

```
device(config)# radius-server host 10.38.37.180 use-vrf mgmt-vrf
device(config-host-10.38.37.180/mgmt-vrf)#
```

- The following examples show how to configure some parameters for communication with the RADIUS server using the mgmt-vrf.

- (Optional) Configure the authentication protocol to use for communication with the RADIUS server.

```
device(config-host-10.38.37.180/mgmt-vrf)# protocol pap
```

- (Optional) Specify a text string to be used as a shared secret between the device and the RADIUS server.

```
device(config-host-10.38.37.180/mgmt-vrf)# key "new#vertigo*secret"
```

- (Optional) Specify the wait time (in seconds) allowed for a RADIUS server response.

```
device(config-host-10.38.37.180/mgmt-vrf)# timeout 10
```

- Return to Privileged EXEC mode.

```
device(config-host-10.38.37.180/mgmt-vrf)# end
```

- Verify the configuration.

```
device# show running-config radius-server host 10.38.37.180

radius-server host 10.38.37.180 use-vrf mgmt-vrf
protocol pap key "60/2cBziRKSGWM6jyUagFdsJ+KICcgECAZGURh0GQSI=\n" encryption-level 7 timeout 10
```

Modifying the RADIUS server configuration

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

```
device# configure terminal
Entering configuration mode terminal
```

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

```
device(config)# radius-server host ?

Possible completions:
  INETADDRESS      Domain name or IP Address of this RADIUS server
```

- Enter the **radius-server host** command with the IP address of the server you want to modify and the **use-vrf** option.

```
device(config)# radius-server host 10.38.37.180 use-vrf mgmt-vrf
```

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

4. Configure the values that you want to change.

- (Optional) The following example shows how to configure a new key.

```
device(config-host-10.38.37.180/mgmt-vrf)# key "changedsec"
```

- (Optional) The following example shows how to configure a timeout value of 3 seconds.

```
device(config-host-10.38.37.180/mgmt-vrf)# timeout 3
```

5. Return to Privileged EXEC mode.

```
device(config-host-10.38.37.180/mgmt-vrf)# end
```

6. **NOTE**

This command does not display default values.

Verify the new configuration.

```
device# show running-config radius-server host 10.38.37.180
radius-server host 10.38.37.180 use-vrf mgmt-vrf
  protocol pap key "h8mcoUf2LZF+P+AjaYn0lQ==\n" encryption-level 7 timeout 3
!
```

NOTE

To remove a server from the list of configured RADIUS servers, use the **no radius-server host** command specifying the IP address or hostname of the RADIUS server that is to be removed.

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 the Login authentication mode section 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. The user records must be synchronized in order to operate. These are the options for creating 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.

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 it to accept two-factor authentication input. When the user logs in, the password or token code works automatically without any changes to the device, as shown in the following example.

```

Welcome to Console Server Management Server

HQ1-4E23-TS1 login: muser34
Password: ***** <-----For example password/8675309

device#

```


TACACS+ Server Authentication

- Understanding and configuring TACACS+ 55

Understanding and configuring TACACS+

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 devices seamlessly integrates into network fabric environments. Once configured to use TACACS+, a device becomes a network access server.

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

TACACS+ authentication, authorization, and accounting

The TACACS+ server is used for authentication, authorization, and accounting. You can access the device through the serial port, or through Telnet or SSH, from either the management interface or the data ports (Ethernet interface or in-band). The device goes through the same TACACS+-based authentication with either access method.

Supported TACACS+ packages and protocols

Extreme 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 device 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 authorization and accounting), as well as configuring TACACS+ on the server. Support for mixed environments may also be required.

Client configuration for TACACS+ support

Each 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 device for AAA service.

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

TABLE 6 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	Specifies the text string that is used as the shared secret between the device and the TACACS+ server to make the message exchange secure. The key must be between 1 and 40 characters in length. The default key is sharedsecret . The exclamation mark (!) is supported both in RADIUS and TACACS+ servers, and you can specify the password in either double quotes or the escape character (\), for example " secret!key " or secret\!key . The only other valid characters are alphanumeric characters (such as a-z and 0-9) and underscores. No other special characters are allowed.
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. Possible values are 0 or 7, where 0 represents store the key in clear-text format and 7 represents encrypted format. Default is 7 (encrypted).
use-vrf	Specifies a VRF through which to communicate with the TACACS+ server.

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.

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. From privileged EXEC mode, enter global configuration mode.

```
device# configure terminal
Entering configuration mode terminal
```

2. Specify the server IP address.

```
device(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 mode where you can specify additional parameters.

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

```
device(config-tacacs-server-fec0:60:69bc:94:211:25ff:fec4:6010)# protocol chap key
"new#hercules*secret"
```


- Return to global configuration mode.

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

- Verify the configuration.

```
device# 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

- From privileged EXEC mode, enter global configuration mode.

```
device# configure terminal
Entering configuration mode terminal
```

- Display the configured server IP addresses.

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

- Enter TACACS+ server configuration mode.

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

- Specify the parameters that you want to modify. This example shows how to modify the shared secret key.

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

- Return to privileged EXEC mode.

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

- Verify the configuration.

```
device# 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!
```

This command does not display default values.

Removing the client-side TACACS+ server configuration

- From privileged EXEC mode, enter global configuration mode.

```
device# configure terminal
Entering configuration mode terminal
```

- Remove a specific TACACS+ server from the list of configured TACACS servers. The following example removes the TACACS+ server fec0:60:69bc:94:211:25ff:fec4:6010.

```
device(config)# no tacacs-server host fec0:60:69bc:94:211:25ff:fec4:6010
```

When authentication, authorization, or accounting mode is set to **tacacs**, you cannot delete from the device configuration, the last server in the list of configured TACACS+ servers; when you attempt to delete the last server on the list, deletion is denied.

Configuring the client to use TACACS+ for login authentication

After you configure the client-side TACACS+ server list, you must set the authentication mode so that TACACS+ is used as the primary source of authentication.

Client configuration for TACACS+ authorization

AAA command authorization is supported for TACACS+

Authorization is the action of determining what a user is allowed to do on a device. By default, TACACS+ command authorization is disabled. Regardless of how authentication is performed (whether it is local, or done on a RADIUS or TACACS+ server), when at least one TACACS+ server is configured, you can enable TACACS+ command authorization by using the **aaa authorization commands** command.

When TACACS+ command authorization is enabled and a device user attempts to execute a command, an authorization request is sent to servers on the TACACS+ server list in a round-robin fashion. In response to the authorization request, the TACACS+ server replies with an accept or reject message based on the user's configuration or settings on the TACACS+ server. When an accept message is received, the user is permitted to execute the command.

At device level, authorization is enforced by Brocade role-based control (RBAC). To ensure that local device-level authorization is done when the TACACS+ server is unreachable, enable command authorization by using the **aaa authorization commands** command specifying the **local** option. When the **local** option is not specified, local device-level authorization is not performed when the TACACS+ server is unreachable; therefore, command authorization fails.

TACACS+ command authorization can only be enabled when at least one TACACS+ server is configured. Similarly, when command authorization is enabled, then the TACACS+ server cannot be removed when it is the only server on the TACACS+ server list.

Limitations

TACACS+ command authorization:

- Is not supported during post boot, or configuration replay
- Only applies to commands executed in global configuration mode; command authorization is not performed for commands executed in sub-configuration mode
- Is not supported by REST API or NetConf

In addition, the commands listed in the following table are not authorized through the TACACS+ server.

TABLE 7 Commands that are not authorized through TACACS+ server

Command type	Command		
Configuration	abort	exit	service
	end	help	top
Operational	cipherset	logout	telnet
	delete	ping	timestamp
	dir	prompt1	oscmd
	devtools	prompt2	show cipherset
	df	python	show cli
	exit	rename	show file
	execute-script	script	show history
	help	send	show netconf-state
	history	reload	show parser dump
	image snapshot	quit	show startup-config

TABLE 7 Commands that are not authorized through TACACS+ server (continued)

Command type	Command		
	log-shell	ssh	traceroute

Enabling command authorization

Before enabling command authorization, at least one TACACS+ server must be configured by using the **tacacs-server** command. In addition, any TACACS+ server configured for TACACS+ authorization must be configured with user rules (to accept or reject commands).

Perform the following steps to enable TACACS+ command authorization.

1. From privileged EXEC mode, enter global configuration mode.

```
device# configure terminal
Entering configuration mode terminal
```

2. Enable command authorization.

```
device(config)# aaa authorization command TACACS+ local
```

This example enables TACACS+ authorization specifying the **local** option. In the event that the TACACS+ server is unreachable or responds with an error, device-level authorization is performed when the **local** option is specified.

NOTE

Supported commands fail and there is no way to recover from this, when **aaa authorization** is configured without specifying the **local** option and the configured TACACS+ servers are not reachable.

3. Return to privileged EXEC mode.

```
device(config)# exit
```

4. Verify the configuration.

```
device# show running-config aaa authorization
aaa authorization tacacs+ local
```

The following example show how to enable and verify TACACS+ command authorization, specifying device-level authorization when the TACACS+ server is unreachable or responds with an error.

```
device# configure terminal
Entering configuration mode terminal
device(config)# aaa authorization command TACACS+ local
device(config)# exit

device# show running-config aaa authorization
aaa authorization tacacs+ local
```

Client configuration for TACACS+ accounting

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 mode, the configuration is applied to all nodes in the fabric.

- 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 or REST are not tracked.

When a TACACS+ server is used for authentication, authorization, or accounting, the device attempts to connect to the first TACACS+ server configured in the list. When the first TACACS+ server cannot be reached, the device attempts to send the packets to the next server on the list.

NOTE

When the first server on the list is reachable again, the device falls back to sending packets to the first server.

If authentication is performed through some other mechanism, such as the device-local database or RADIUS, 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, 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 a 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 TACACS+. 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, you cannot remove a TACACS+ server if it is the only server in the list.

Enabling login accounting

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

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

```
device# configure terminal
Entering configuration mode terminal
```

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

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

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

```
device(config)# exit
```

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

```
device(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 device where login accounting is enabled and command accounting is disabled.

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

```
device# configure terminal
Entering configuration mode terminal
```

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

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

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

```
device(config)# exit
```

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

```
device# 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 global configuration mode.

The following examples show two ways of disabling command accounting. The commands are executed in global configuration mode.

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

The following examples show two ways of disabling login accounting.

```
device(config)# aaa accounting exec default start-stop none
device(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.

The following 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 depends 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 4/9/2016 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 5/14/2016 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 5/14/2016 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 your server manufacturer. 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 device. Along with each account name, you must assign appropriate device access roles. A user account can exist on TACACS+ servers 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 an Extreme 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.

Establishing a server-side user account

The following example assigns the user "Mary" the Extreme role of "vlanadmin" and different passwords depending on whether CHAP or PAP is used. In the following example, which works in an environment with only devices supported by this guide, the `brcd-role` attribute is mandatory. In a mixed-vendor environment, the `brcd-role` attribute must be set to optional. Refer to [Configuring TACACS+ for a mixed-vendor environment](#) on page 65 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. The following example allows the user "fred" to be authenticated using the `/etc/passwd` file.

```
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 Extreme role concept, has no relation with the value of the "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 "Extreme" with the group "admin".

```
user = Extreme {
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 = Extreme {
member = admin
expires = "Jan 01 2017"
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 device and the TACACS+ server. The TACACS+ server key must be configured on both the TACACS+ server and the client device. Only one key is defined per server in the TACACS+ server configuration file. The key is defined as follows:

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

Configuring TACACS+ for the AAA user role

Configuring TACACS+ for the AAA user role allows the AAA user role to access configuration commands.

At least one TACACS+ server must be configured on the device using the **tacacs-server host** command.

You must configure a server-side user role on the TACACS+ server. Refer to [Configuring TACACS+ for a mixed-vendor environment](#) on page 65 for more information.

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

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

Configuring server-side rules for TACACS+ command authorization

To perform TACACS+ command authorization, a TACACS+ server must be configured with user rules to accept or reject commands.

The following example shows a rule configuration for a user named **tacuser**. In this configuration, a reject message is returned for the **show vrf** command and an accept message is returned for all other **show** commands.

```
user = tacuser {
  default service = permit
  chap = cleartext "password"
  service = exec {
    brcd-role = admin
  }
  cmd = show {
    deny vrf
    permit .*
  }
}
```


Configuring TACACS+ for a mixed-vendor environment

Extreme 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 platforms supported by this guide and other platforms. You can use TACACS+ to provide centralized AAA services to multiple network access servers or clients. To use TACACS+ services in mixed-vendor environments, you must configure the Attribute-Value Pair (AVP) argument to be optional, as shown in the following example.

```
brcd-role*admin
```

The device sends the optional argument **brcd-role** in the authorization request to the TACACS+ server. 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 Extreme users are able to successfully authorize with all TACACS+ servers in a mixed-vendor environment.

Configuring optional arguments in *tac_plus*

The following example is specific to the *tac_plus* package. The syntax for other packages may differ.

In the example, the mandatory attribute `priv-lvl=15` is set to allow the server to authenticate. The optional `brcd-role = admin` argument is added to the *tac_plus.conf* file and allows devices to authenticate.

The following example configures a user with the optional AVP, `brcd-role = admin`. An Extreme 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>
}
```


HTTPS Certificates

- [HTTPS certificate overview.....](#) 67
- [Configuring HTTPS certificates.....](#) 67
- [Disabling HTTPS certificates.....](#) 69
- [Enabling HTTPS service.....](#) 70
- [Disabling HTTPS service.....](#) 71
- [Importing TLS certificate and keys without trust point.....](#) 71

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.

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 device 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 configure terminal mode.

```
device#configure terminal
```

2. Enter RBridge ID mode.

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

3. 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.

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

4. 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.

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

- 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.

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

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

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

- You must authenticate the 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.

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

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

```
device# 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: *****
```

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

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

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

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

```
device# show crypto ca certificates
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
```

```
device# 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 an enabled state (by default HTTP is enabled), then execute the **http server** command to shutdown the service, followed by **no http server** command to enable HTTPS.
 - If HTTP is in a disabled state, then execute the **no http server** command to enable HTTPS.
 - Reboot the device.
 - 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 device certificate with the **no crypto ca import** command.

```
device# no crypto ca import t1 certificate

device# 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.

```
device# no crypto ca authenticate t1
device# show crypto ca certificates
rbridge-id:1
Trustpoint: t1
certificate: none
CA certificate: none
```

3. Enter configure terminal mode.

```
device#configure terminal
```

4. Enter RBridge ID mode.

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

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

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

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

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

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

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

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

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

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

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. Refer to the *Extreme Network OS Command Reference*.
- Reboot the entire device.
- Commit an HA failover, if that option is available.

Disabling HTTPS service

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

Refer to the *Extreme Network OS Command Reference*.

Importing TLS certificate and keys without trust point

This feature allows TLS server certificates (third party CA certificate) and keys to be directly imported without any trust point support.

The SSL/TLS protocol uses a pair of keys – one private, one public – to authenticate, secure and manage secure connections. These keys are created together as a pair and work together during the TLS handshake process to set up a secure session.

As part of the TLS handshake, the protocol also allows both peers to authenticate their identity. Finally, with encryption and authentication in place, the TLS protocol also provides its own message framing mechanism and signs each message with a message authentication code (MAC). Combined, all three mechanisms serve as a foundation for secure communication.

You must create a username named "scpuser" with admin privileges before you import the certificate or key.

Both the certificate and the key is used to establish secure connection over TLS by restarting the http server with the **http server use-vrf <VRF name> shutdown** and **no http server use-vrf <VRF name> shutdown** commands.

1. Enter configure terminal mode.

```
device#configure terminal
```

2. Enter RBridge ID mode.

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

3. Create a user account called "scpuser" that is assigned the admin role, using the **username** command.

```
device(config)# username scpuser role admin password 123456
```

4. Import the certificate file using the standard Linux **scp** command from any Linux machine.

```
# scp certificatefile scpuser@10.16.0.112:flash-tlscert
```

5. Import the key file using the standard Linux **scp** command from any Linux machine.

```
# scp keyfile scpuser@10.16.0.112:flash-tlsprivkey
```

6. Reboot the HTTP service on the device with the **http server use-vrf <VRF name> shutdown** and **no http server use-vrf <VRF name> shutdown** commands.

```
device(config)# http server use-vrf myvrf shutdown
device(config)# no http server use-vrf myvrf shutdown
```

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

```

device# show http server status
rbridge-id 1:
VRF-Name: mgmt-vrf           Status: HTTP Enabled and HTTPS Disabled
VRF-Name: default-vrf       Status: HTTP Enabled and HTTPS Disabled

device# show http server status
rbridge-id 1:
VRF-Name: mgmt-vrf           Status: HTTP Disabled and HTTPS Enabled
VRF-Name: default-vrf       Status: HTTP Enabled and HTTPS Disabled

device# show cert-util tlsprivkey
%%Info: RSA Private key is already installed on the device.

device# show cert-util tlscert
Displaying contents of tlscert.pem
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 8209 (0x2011)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: C=IN, ST=Karnataka, L=Bangalore, O=Brocade, OU=NI, CN=Brocade/
    emailAddress=gperakam@brocade.com
    Validity
      Not Before: Oct  3 05:26:25 2017 GMT
      Not After : Oct 13 05:26:25 2018 GMT
    Subject: C=US, ST=CA, L=SJ, O=Brocade, OU=Eng, CN=brocade
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      Public-Key: (2048 bit)
      Modulus:
        00:a2:34:c7:52:13:61:32:70:79:62:da:65:be:c6:
        46:ff:3c:e7:04:c8:c4:73:e7:47:62:91:9a:77:48:
        db:ab:43:ea:23:e8:97:b4:3f:95:f1:cf:7b:7a:8a:
        4c:e2:2c:26:fe:17:5e:14:d2:e9:cc:3b:39:2d:65:
        f7:80:dc:45:c0:24:d2:40:3e:71:6b:4f:22:ef:cd:
        c8:ac:00:c1:14:ff:e6:54:98:17:46:a5:0f:d6:17:
        7e:18:6f:fd:5d:e9:67:6b:fa:dd:3d:df:26:08:46:
        3b:4b:ba:38:ed:43:f0:d8:d9:db:62:1a:17:c4:5a:
        6e:d6:ac:4b:e4:3d:06:ae:61:8f:e4:fa:63:27:08:
        48:27:39:86:24:cf:f9:26:2c:6e:07:f5:0a:4e:d7:
        4f:ff:b9:c8:f2:93:96:b8:3d:5f:d5:63:4e:3d:1f:
        44:f2:c9:f6:3a:cf:12:00:fc:fb:cc:b3:d9:3a:7f:
        92:ab:e5:f2:47:b0:1e:3e:6e:da:e0:c5:dd:88:38:
        89:93:8c:75:af:8e:e5:10:6e:47:98:d9:86:81:8c:
        3d:1d:a1:b5:78:99:48:4e:49:e8:4a:7e:b8:21:07:
        c5:00:5f:3f:44:61:d3:85:44:e3:20:21:45:68:dd:
        64:db:4b:70:98:c5:f4:53:86:e4:27:40:67:a1:3b:
        1b:79
      Exponent: 65537 (0x10001)
    X509v3 extensions:
      X509v3 Basic Constraints:
        CA:FALSE
      Netscape Cert Type:
        SSL Server
      Netscape Comment:
        OpenSSL Generated Server Certificate
      X509v3 Subject Key Identifier:
        66:EB:24:45:03:05:7D:A6:9D:30:77:E0:07:5A:A7:24:DA:8A:1D:7A
      X509v3 Authority Key Identifier:
        keyid:1F:7D:8D:B0:DB:BA:F6:41:8F:8C:6B:85:55:C6:4B:C2:54:3A:77:80
        DirName:/C=IN/ST=Karnataka/L=Bangalore/O=Brocade/OU=NI/CN=Brocade/
        emailAddress=gperakam@brocade.com
        serial:DF:A7:C9:93:BF:C9:23:37

      X509v3 Key Usage: critical
        Digital Signature, Key Encipherment
      X509v3 Extended Key Usage:
        TLS Web Server Authentication
    Signature Algorithm: sha256WithRSAEncryption

```



```
b4:b4:af:9f:18:aa:d0:82:2c:15:0f:e8:5f:2b:5a:65:6e:2e:
b3:be:57:9f:b1:4a:e6:21:27:20:8b:e2:dc:66:99:98:5a:35:
32:8b:72:4c:2a:29:62:d6:a3:11:4c:bb:46:65:71:de:ac:45:
57:e7:c0:a5:51:80:04:a0:63:9d:26:bc:86:8f:df:86:d5:fa:
1b:b3:ad:bc:3d:ce:23:2f:4a:05:51:6b:c0:45:f0:90:73:fa:
70:7c:7f:5e:50:a2:bd:d8:48:d6:85:08:c2:e0:c7:b7:dd:75:
55:fa:11:c9:e9:6e:c2:db:01:c0:60:f0:63:2a:ee:95:22:f9:
f8:e8:44:9b:d4:02:b5:66:7a:aa:44:9d:5c:08:d8:c7:1f:23:
46:bc:e9:8b:d6:08:23:f5:c5:68:68:b1:bd:96:ac:c4:cc:4a:
25:36:34:95:0c:c6:c0:04:ca:d6:8e:31:f5:9c:0e:1a:3d:b4:
7d:4f:3c:c0:dd:47:5b:b5:1f:74:41:49:59:f8:dd:7d:a6:7a:
50:1e:aa:d0:49:77:f8:bc:10:91:cf:90:12:28:df:72:f2:f0:
fb:d6:df:da:6e:1f:c8:65:99:e5:07:4a:b6:dd:db:1c:b0:33:
18:64:a2:b6:f2:ef:84:62:24:07:84:2d:38:ba:6e:58:fe:98:
df:c6:0f:f4
```


ACLs

• ACL overview.....	75
• Layer 2 (MAC) ACLs.....	78
• Layer 3 (IPv4 and IPv6) ACLs.....	84
• ACL Show and Clear commands.....	96

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, deny, or hard-drop traffic according to source address only.
- *Extended ACLs* — Permit, deny, or hard-drop 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/UDP port name or number
 - 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 8 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 78 Layer 3 (IPv4 and IPv6) ACLs on page 84
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 78 Layer 3 (IPv4 and IPv6) ACLs on page 84 <i>Extreme Network OS Layer 2 Switching Configuration Guide > "VXLAN Overlay Gateways for NSX Controller Deployments"</i>
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 84

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 9 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	<i>Extreme Network OS Management Configuration Guide > "SNMP" > "Managing SNMP access rights using ACLs"</i>
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.	<i>Extreme Network OS Security Configuration Guide > "Configuring Dynamic ARP Inspection (DAI)" > "Implementing ARP ACLs for DAI"</i>

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 84.

Otherwise, continue with [ACLs applied to interfaces](#) on page 77.

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 10 ACL and rule limits

Resource	VDX 6740 VDX 6940 VDX 2741 VDX 2746	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

TABLE 10 ACL and rule limits (continued)

Resource	VDX 6740 VDX 6940 VDX 2741 VDX 2746	VDX 8770
Maximum total IPv4 ACLs (standard and extended)	512	2048
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

NOTE

The hardware profile configured on the VDX device defines the number of supported ACLs. Consult the Release Notes for the ACL limits for each hardware profile.

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

Layer 2 access control lists (ACLs) filter traffic based on MAC header fields.

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 terminal** to access global configuration mode.

```
device# configure terminal
```

2. Enter the **mac access-list standard** command to create the ACL.

```
device(config)# mac access-list standard test_01
device(config-macl-std) #
```

3. For each ACL rule that you need to create, enter a permit or deny command, specifying the needed parameters.

```
device(config-macl-std) # seq 100 deny host 0011.2222.3333 count
device(config-macl-std) # seq 110 permit host 0022.1111.2222 ffff.ffff.00ff count
device(config-macl-std) # deny host 0022.3333.4444 count
device(config-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 terminal** command to access global configuration mode.

```
device# configure terminal
```


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 terminal** command to access global configuration mode.

```
device# configure terminal
```

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 terminal** command to access global configuration mode.

```
device# configure terminal
```

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 MAC ACL rules enabled for counter statistics

When you create ACL rules, the **count** parameter enables you to display counter statistics.

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
```

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-macl-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 terminal** command to access global configuration mode.

```
device# configure terminal
```

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 76.

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 85	
	One of the following procedures: <ul style="list-style-type: none"> • Creating a standard IPv4 ACL on page 87 • Creating a standard IPv6 ACL on page 88 • Creating an extended IPv4 ACL on page 88 • Creating an extended IPv6 ACL on page 89 	
Applying Layer 3 ACLs to interfaces on page 89		Applying Layer 3 rACLs to RxBridges on page 92

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 93
- [Reordering the sequence numbers in a Layer 3 ACL](#) on page 93
- [ACL counter statistics \(Layer 3\)](#) on page 94
- [ACL logs](#) on page 83
- [ACL Show and Clear commands](#) on page 96

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 device, 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 device 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 device 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

The following protection guards against malicious ICMP timestamp requests (icmp-type 13):

- ICMP timestamp requests and responses are dropped by default.
- If an ACL with a **permit icmp any any** rule is applied to a management interface, such a rule permits ICMP timestamp requests. However, ICMP timestamp responses are blocked.

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.

```
device# configure
```

2. Enter the **ip access-list standard** command to create the access list.

```
device(config)# ip access-list standard stdACL3
```

3. For each ACL rule, enter a **seq** command, specifying the needed parameters.

```
device(config-ipacl-std)# seq 5 permit host 10.20.33.4
device(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.

```
device# configure
device(config)# ip access-list standard stdACL3
device(config-ipacl-std)# seq 5 permit host 10.20.33.4
device(config-ipacl-std)# seq 15 deny any
device(config-ipacl-std)# exit
device(config)# interface tengigabitethernet 122/5/22
device(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.

```
device# configure
```

2. Enter the **ipv6 access-list standard** command to create the access list.

```
device(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.

```
device(config-ipv6acl-std)# seq 5 permit host 2001:db8::1:2
device(config-ipv6acl-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.

```
device# configure
```

2. Enter the **ip access-list extended** command to create the access list.

```
device(config)# ip access-list extended extdACL5
```

3. For each ACL rule, enter a **[seq] {permit | deny | hard-drop}** command—specifying the needed parameters.

```
device(config-ipacl-ext)# seq 5 deny tcp host 10.24.26.145 any eq 23
device(config-ipacl-ext)# seq 7 deny tcp any any eq 80
device(config-ipacl-ext)# seq 10 deny udp any any range 10 25
device(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.

```
device# configure
```

2. Enter the **ipv6 access-list extended** command to create the access list.

```
device(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.

```
device(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.

```
device# configure
device(config)# ipv6 access-list extended ip_acl_1
device(conf-ip6acl-ext)# seq 10 deny ipv6 any any dscp 3
device(conf-ip6acl-ext)# seq 20 deny ipv6 2001:2002:1234:1::/64 2001:1001:1234:1::/64 count
device(conf-ip6acl-ext)# exit
device(config)# interface ten 122/5/22
device(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 **access-group** command.

1. Enter **configure** to change to global configuration mode.

```
device# configure
```

2. Enter the **interface** command, specifying the interface type and the rbridge-id/slot/port number.

```
device(config)# interface ten 122/5/22
```

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(conf-if-te-122/5/22)# ipv6 access-group ip_acl_1 in
```

4. (Optional) To display updated ACL details, enter the **show access-list** command.

```
device(conf-if-te-122/5/22)# 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)
```

The following example shows how to apply an IPv6 ACL to a physical interface.

```
device# configure
device(config)# interface ten 122/5/22
device(conf-if-te-122/5/22)# ipv6 access-group ip_acl_1 in

device(conf-if-te-122/5/22)# 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 terminal** command to access global configuration mode.

```
device# configure terminal
```

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 (attached to a VLAN).

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
```

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**.

```
device(config-overlay-gw-gw121)# ip access-group stdipaclin in
device(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 VCS mode, you can apply an ACL to any fabric 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.

```
device# configure
```

2. Use the **interface management** command to enter configuration mode for the management interface, specifying RBridge ID/port.

```
device(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**.

```
device(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**.

```
device(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.

```
device(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.

```
device# configure
```

2. Enter the **interface** command, specifying the interface type and name.

```
device(config)# interface tengigabitethernet 122/5/22
```

3. Enter the **no access-group** command.

```
device(config-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(config-ippacl-ext)# hard-drop tcp host 10::1 any count
device(config-ippacl-ext)# hard-drop udp any host 20::1 count
device(config-ippacl-ext)# permit tcp host 10::2 any eq telnet count
device(config-ippacl-ext)# permit tcp host 10::2 any eq bgp count
device(config-ippacl-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 terminal** command to access global configuration mode.

```
device# configure terminal
```

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.

```
device# 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.

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 terminal** to access global configuration mode.

```
device# configure terminal
```

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 terminal** command to access global configuration mode.

```
device# configure terminal
```

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.

The following example shows how to create an IPv6 extended ACL and define a counter-enabled rule for it.

```
device# configure terminal
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.

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 11 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 specific ACL or only for that ACL on a specific interface or RBridge. You can also display information for all ACLs bound to a specific device 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 specific 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 specific ACL or only for that ACL on a specific interface or Rbridge. You can also display statistical information for all ACLs bound to an interface, RBridge, or VXLAN overlay gateway.

TABLE 12 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 specific ACL or only for that ACL on a specific interface or RBridge. You can also clear statistical information for all ACLs bound to a specific interface, RBridge, or VXLAN overlay gateway.

PBR - Policy-Based Routing

- Policy-Based Routing..... 97
- Policy-Based Routing behavior..... 98
- Policy-Based Routing with differing next hops..... 99
- Policy-Based Routing uses of NULL0..... 100

Policy-Based Routing

Policy-Based Routing (PBR) allows you to use ACLs and route maps to selectively modify and route IP packets in hardware.

Basically, the ACLs classify the traffic and route maps that match on the ACLs set routing attributes for the traffic. A PBR policy specifies the next hop for traffic that matches the policy, as follows:

- For standard ACLs with PBR, you can route IP packets based on their source IP address.
- For extended ACLs with PBR, you can route IP packets based on all of the matching criteria in the extended ACL.

NOTE

For details about ACLs, refer to the "ACLs" section of the *Network OS Security Configuration Guide*.

To configure PBR, you define the policies using IP ACLs and route maps, then enable PBR on individual interfaces. The platform programs the ACLs on the interfaces, and routes traffic that matches the ACLs according to the instructions provided by the "set" statements in the route map entry.

Currently, the following platforms support PBR:

- VDX 8770
- VDX 6740
- VDX 6740T
- VDX 6740T-1G
- VDX 6940

You can configure the device to perform the following types of PBR based on a packet's Layer 3 and Layer 4 information:

- Select the next-hop gateway.
- Set the DSCP value.
- Send the packet to the null interface (null0) to drop the packets.

Using PBR, you can define a set of classifications that, when met, cause a packet to be forwarded to a predetermined next-hop interface, bypassing the path determined by normal routing. You can define multiple match and next-hop specifications on the same interface. The configuration of a set of match criteria and corresponding routing information (for example next hops and DSCP values) is referred to as a stanza.

You can create multiple stanzas within a route-map configuration and assign the stanza an "Instance_ID" that controls the program positioning within the route map. Furthermore, when the route map is created, you specify a deny or permit construct for the stanza. In addition, the ACL used for the "match" criteria also contains a deny or permit construct.

The deny or permit nomenclature has a different meaning within the context of the PBR operation than it does within the normal context of user-applied ACLs (where deny and permit are directly correlated to the forwarding actions of forward and drop). The following table

lists the behavior between the permit and deny actions specified at the route-map level, in conjunction with the permit and deny actions specified at the ACL rule level.

Route-map level permit and deny actions	ACL clause permit and deny actions	Resulting Ternary Content Addressable Memory (TCAM) action
Permit	Permit	The "set" statement of the route-map entry is applied.
Permit	Deny	The packet is "passed" and routed normally. The contents of the "set" command are not applied. A rule is programmed in the TCAM as a "permit" with no result actions preventing any further statements of the route-map ACL from being applied.
Deny	Permit	The packet is "passed" and routed normally. There should be no "set" commands following the "match" command of a deny route-map stanza. A rule is programmed in the TCAM as a "permit" with no result actions preventing any further statements of the route-map ACL from being applied.
Deny	Deny	No TCAM entry is provisioned; no other route-map ACL entries will be compared against. If no subsequent matches are made, the packet is forwarded as normal.

Notes:

- Ternary Content Addressable Memory is high-speed hardware memory.
- Consider the permit and deny keywords as allowing the specified match content as either being permitted to or denied from using the defined "set criteria" of the route map. The permit and deny keywords do not correlate to the forwarding action of forward and drop as they do in the ACL application.
- PBR route maps may only be applied to Layer 3 (L3) interfaces. Application of a route map to a non-L3 interface results in the configuration being rejected.
- Deletion of a route map or deletion of an ACL used in the route map "match" is not allowed when the route map is actively bound to an interface. Attempts to delete an active route map or associated ACL is rejected, and an error and log will be generated.
- The "set" commands are only available within the context of a "permit" stanza. The CLI should not allow the use of a "set" command within a PBR "deny" stanza.

Policy-Based Routing behavior

Policy-Based Routing (PBR) next-hop behavior selects the first live next-hop specified in the policy that is "UP".

If none of the policy's direct routes or next hops is available, the packets are forwarded as per the routing table. The order in which the next hop addresses are listed in the route map is an implicit preference for next hop selection. For example, if you enter the next hop addresses A, B, and C (in that order), and all paths are reachable, then A is the preferred selection. If A is not reachable, the next hop is B. If the path to A becomes reachable, the next hop logic will switch to next-hop A.

PBR does not have implicit "deny ip any any" ACL rule entry, as used in ACLs, to ensure that for route maps that use multiple ACLs (stanzas), the traffic is compared to all ACLs. However, if an explicit "deny ip any any" is configured, traffic matching this clause is routed normally using L3 paths and is not compared to any ACL clauses that follow the clause.

The set clauses are evaluated in the following order:

1. Set clauses where the next hop is specified.
2. Set interface NULL0.

The order in which you enter either the **set ip next-hop** or the **set ipv6 next-hop** command determines the order preference. If no next-hops are reachable, the egress interface is selected based on the order of interface configuration. The set interface NULL0 clause — regardless of which position it was entered — is always placed as the last selection in the list.

For example if you enter the order shown below, the PBR logic will treat 3.3.3.5 as its first choice. If 3.3.3.5 is unavailable, the PBR logic will determine if 6.6.6.7 is available. NULL0 is recognized only if 3.3.3.5 and 6.6.6.7 are both unavailable.

```
route-map foo permit 20
  match ip address acl Vincent
  set ip next-hop 3.3.3.5
  set ip interface NULL0
  set ip next-hop 6.6.6.7
```

NOTE

If a PBR route map is applied to an interface that is actively participating in a control protocol, and the ACL specified in the route map also matches the control protocol traffic, the control protocol traffic is trapped to the local processor and is not forwarded according to the route map.

Policy-Based Routing with differing next hops

In this example, traffic is routed from different sources to different places (next hops). Packets arriving from source 1.1.1.1 are sent to the VRF pulp_fiction's next hop at 3.3.3.3; packets arriving from source 2.2.2.2 are sent to the VRF pulp_fiction's next hop at 3.3.3.5. If next hop 3.3.3.5 is not available, then the packet is sent to the next hop 2001:db8:0:0:ff00:42:8329.

1. Configure the ACLs.

```
device(config)# ip access-list standard Jules
device(conf-ipacl-std)# permit ip 1.1.1.1

device(config)# ip access-list standard Vincent
device(conf-ipacl-std)# permit ip 2.2.2.2
```

2. Create the first stanza of the route map, which is done in RBridge ID configuration mode. (The example is using a route-map named pulp_fiction.)

```
with(config)# rbridge-id 1
device(config-rbridge-id-1)# route-map pulp_fiction permit 10
device(config-routemap pulp_fiction)# match ip address acl Jules
device(config-routemap pulp_fiction)# set ip vrf pulp_fiction next-hop 3.3.3.3
```

3. Create the second stanza of the route-map (in this example we'll define a route-map named pulp_fiction.)

```
device(config-rbridge-id-1)# route-map pulp_fiction permit 20
device(config-routemap pulp_fiction)# match ip address acl Vincent
device(config-routemap pulp_fiction)# set ip vrf pulp_fiction next-hop 3.3.3.5
device(config-routemap pulp_fiction)# set ip next-hop 6.6.6.7
```

4. Bind the route map to the desired interface.

```
device(config)# interface TenGigabitEthernet 4/1
device(conf-if-te-4/1)# ip policy route-map pulp_fiction
```

5. View the route map configuration contents.

```
device# show running-config route-map pulp-fiction
route-map pulp-fiction permit 10
match ip address acl Jules
  set ip vrf pulp_fiction next-hop 3.3.3.3
!
route-map pulp-fiction permit 20
match ip address acl Vincent
  set ip vrf pulp_fiction next-hop 3.3.3.5
  set ip next-hop 6.6.6.7
!
```

6. View the route map application.

```
device# show route-map pulp-fiction
Interface TenGigabitEthernet 3/3
  route-map pulp-fiction permit 10
    match ip address acl Jules      (Active)
    set ip vrf pulp_fiction next-hop 3.3.3.3
    Policy routing matches: 0 packets; 0 bytes

  route-map pulp-fiction permit 20
    match ip address acl Vincent    (Active)
    set ip vrf pulp_fiction next-hop 3.3.3.5 (selected)
    set ip next-hop 6.6.6.7
    Policy routing matches: 0 packets; 0 bytes
```

NOTE

For the first stanza (10) created in step 2, the absence of the keyword selected indicates that the none of the next hops in the list is being used; the packet is being routed by the standard routing mechanism.

Policy-Based Routing uses of NULL0

NULL0 is a mechanism used to drop packets in policy-based routing.

NULL0 is a mechanism used to drop packets in policy-based routing. If the NULL0 interface is specified within a stanza and the stanza also contains a “match ACL” statement, only traffic meeting the match criteria within the ACL is forwarded to the NULL0 interface. If the NULL0 interface is specified within a stanza that does not contain a “match” statement, the match criteria is implicitly “match any.”

Examples of using NULL0 include:

- NULL0 in conjunction with a “match” statement.
- NULL0 as a default action of a route map.

Policy-Based Routing and NULL0 with match statements

NULL0 is a mechanism used to drop packets in the Policy-Based Routing (PBR). If the NULL0 interface is specified within a stanza and the stanza also contains a “match ACL” statement, only traffic meeting the match criteria within the ACL is forwarded to the NULL0 interface. If the NULL0 interface is specified within a stanza that does not contain a “match” statement, the match criteria is implicitly “match any.”

In this example, the use of the NULL0 interface is only applicable to frames that meet the match criteria defined in the created ACL, or implicit "permit any" when no explicit match statement is listed for the stanza.

1. Configure the ACLs.

```
sw0(config)# ip access-list standard Jules
sw0(conf-ipacl-std)# permit ip 1.1.1.1
sw0(conf-ipacl-std)# deny ip 11.11.11.11
sw0(config)# ip access-list standard Vincent
sw0(conf-ipacl-std)# permit ip 2.2.2.2
```

2. Create the first stanza of the route map, which is done in RBridge ID configuration mode. (The example is using a route-map named pulp_fiction.)

```
sw0(config)# rbridge-id 1
sw0(config-rbridge-id-1)# route-map pulp_fiction permit 10
sw0(config-routemap pulp_fiction)# match ip address acl Jules
sw0(config-routemap pulp_fiction)# set ip vrf pulp_fiction next-hop 3.3.3.3
sw0(config-routemap pulp_fiction)# set ip interface NULL0
```

3. Create the second stanza of the route map. (The example is using a route map named pulp_fiction.)

```
sw0(config-rbridge-id-1)# route-map pulp_fiction permit 20
sw0(config-routemap pulp_fiction)# match ip address acl Vincent
sw0(config-routemap pulp_fiction)# set ip vrf pulp_fiction next-hop 3.3.3.5
sw0(config-routemap pulp_fiction)# set ipv6 next-hop 2001:db8:0:0:0:ff00:42:8329
```

Based on the above configuration, when address 1.1.1.1 is received, it matches stanza 10:

- If the next hop 3.3.3.3 is selected, the packet is forwarded to 3.3.3.3.
- If 3.3.3.3 is not selected by the PBR logic, the packet is sent to the next specified next-hop, which is the NULL0 interface, resulting in the traffic being dropped.
- If address 11.11.11.11 is received, since it matches the deny case of the ACL, it is denied from using the next hops specified in the route map and is forwarded according to the standard logic.
- If address 12.12.12.12 is received, because it meets none of the specified match criteria in either of the two stanzas, it basically falls off the end of the route map and reverts to using the standard routing logic.

Policy-Based Routing and NULL0 as route map default action

This example shows the use of the NULL0 interface.

In this example, the use of the NULL0 interface is only applicable to frames that meet the match criteria defined in the created ACL.

1. Configure the ACLs.

```
sw0(config)# ip access-list standard Jules
sw0(conf-ipacl-std)# permit ip 1.1.1.1
sw0(conf-ipacl-std)# deny ip 11.11.11.11
sw0(config)# ip access-list standard Vincent
sw0(conf-ipacl-std)# permit ip 2.2.2.2
```

2. Create the first stanza of the route map, which is done in RBridge ID configuration mode. (The example is using a route-map named pulp_fiction.)

```
sw0(config)# rbridge-id 1
sw0(config-rbridge-id-1)# route-map pulp_fiction permit 10
sw0(config-routemap pulp_fiction)# match ip address acl Jules
sw0(config-routemap pulp_fiction)# set ip vrf pulp_fiction next-hop 3.3.3.3
sw0(config-routemap pulp_fiction)# set ip interface NULL0
```

3. Create the second stanza of the route map. (The example is using a route-map named pulp_fiction.)

```
sw0(config-rbridge-id-1)# route-map pulp_fiction permit 20
sw0(config-routemap pulp_fiction)# match ip address acl Vincent
sw0(config-routemap pulp_fiction)# set ip vrf pulp_fiction next-hop 3.3.3.5
```

4. Create the third stanza, which provides the default action of the route map.

```
sw0(config-rbridge-id-1)# route-map pulp_fiction permit 30
sw0(config-routemap pulp_fiction)# set ip interface NULL0
```

The above configuration introduces a third stanza that defines the routing desired for all frames that do not meet any of the match criteria defined by the route map.

Based on the above configuration, when address 1.1.1.1 is received, it matches stanza 10:

- If the next hop 3.3.3.3 is selected, the packet is forwarded to 3.3.3.3.
- If 3.3.3.3 is not selected by the PBR logic, the packet is sent to the next specified next-hop, which is the NULL0 interface, resulting in the traffic being dropped.
- If address 11.11.11.11 is received, since it matches the deny case of the ACL, it is denied from using the next hops specified in the route map and will be forwarded according to the standard logic.
- If address 12.12.12.12 is received, because it meets none of the specified match criteria in either of the first two stanzas, it reaches the third stanza. Since a no "match" statement is specified, it is an implicit "match any." The address 12.12.12.12 is forwarded to the NULL0 interface where it is dropped.

Providing the default stanza enables a mechanism whereby if any packet is received that does not meet the match criteria set by the route map, the traffic is dropped.

802.1x Port Authentication

- 802.1x protocol overview..... 103
- Configuring 802.1x authentication..... 103
- MAC authentication 108
- Configuring MAC authentication bypass..... 110
- Configuring MAC authentication..... 112

802.1x protocol overview

The 802.1x protocol defines a port-based authentication algorithm involving network data communication between client-based supplicant software, an authentication database on a server, and the authenticator device. In this situation the authenticator device is the VDX hardware.

As the authenticator, the VDX hardware prevents unauthorized network access. Upon detection of the new supplicant, the VDX hardware enables the port and marks it "unauthorized." In this state, only 802.1x traffic is allowed. All other traffic (for example, DHCP and HTTP) is blocked. The VDX hardware transmits an Extensible Authentication Protocol (EAP) Request to the supplicant, which responds with the EAP Response packet. The VDX hardware then forwards the EAP Response packet to the RADIUS authentication server. If the credentials are validated by the RADIUS server database, the supplicant may access the protected network resources.

When the supplicant logs off, it sends an EAP Logoff message to the VDX hardware, which then sets the port back to the "unauthorized" state.

NOTE

802.1x port authentication is not supported by LAG (Link Aggregation Group) or interfaces that participate in a LAG.

NOTE

The EAP-MD5, EAP-TLS, EAP-TTLS and PEAP-v0 protocols are supported by the RADIUS server and are transparent to the authenticator device.

Configuring 802.1x authentication

The tasks in this section describe the common 802.1x operations that you will need to perform. For a complete description of all the available 802.1x CLI commands for the VDX hardware, refer to the *Extreme Network OS Command Reference*.

Understanding 802.1x configuration guidelines and restrictions

When configuring 802.1x, be aware of this 802.1x configuration guideline and restriction: If you globally disable 802.1x, then all interface ports with 802.1x authentication enabled automatically switch to force-authorized port-control mode.

Configuring authentication

The **radius-server** command attempts to connect to the first RADIUS server. If the RADIUS server is not reachable, the next RADIUS server is contacted. However, if the RADIUS server is contacted and the authentication fails, the authentication process does not check for the next server in the sequence.

Perform the following steps to configure authentication.

1. Enter the **configure** command to change to global configuration mode.

```
device# configure
```

2. Use the **radius-server** command to add RADIUS to the device as the authentication server. This command can be repeated for additional servers. However, this command moves the new RADIUS server to the top of the access list.

```
device(config)# radius-server host 10.0.0.5
```

3. Enable 802.1x authentication globally

```
device(config)# dot1x enable
```

4. Use the **interface** command to select the interface port to modify.

```
device(config)# interface tengigabitethernet 5/1/12
```

5. Use the **dot1x authentication** command to enable 802.1x authentication.

```
device(conf-if-te-5/1/12)# dot1x authentication
```

6. Return to privileged EXEC mode.

```
device(conf-if-te-5/1/12)# end
```

Configuring interface-specific administrative features for 802.1x

It is essential to configure the 802.1x port authentication protocol globally on the VDX hardware, and then enable 802.1x and make customized changes for each interface port. Because 802.1x is enabled and configured in [Configuring 802.1x authentication](#) on page 103, use the administrative tasks in this section to make any necessary customizations to specific interface port settings.

802.1x readiness check

The 802.1X readiness check audits all the ports for 802.1X activity and displays information about the devices with 802.1X-supported ports. The 802.1X readiness check can be used to establish whether the devices connected to the ports are 802.1X-capable.

The 802.1X readiness check is allowed on all ports that can be configured for 802.1X. The 802.1X readiness check is not available on a port that is configured by the **dot1x port-control force-unauthorized** command.

When you execute the **dot1x test eapol-capable** command on an 802.1X-enabled port, and the link comes up, the port queries the connected client about its 802.1X capability. When the client responds with a notification packet, it is 802.1X-capable. A RASLog message is generated if the client responds within the timeout period. If the client does not respond to the query, the client is not 802.1X-capable, and a syslog message is generated indicating the client is not EAPOL-capable.

Follow these guidelines to enable the 802.1X readiness check on the device:

- The 802.1X readiness check is typically used before 802.1X is enabled on the device.
- 802.1X authentication cannot be initiated while the 802.1X readiness check is in progress.
- The 802.1X readiness check cannot be initiated while 802.1X authentication is active.
- 802.1X readiness can be checked on a per-interface basis.
- The 802.1X readiness check for all interfaces at once is not supported.
- The 802.1X test timeout is shown in the output of the **show dot1x** command.

Configuring 802.1x port authentication on specific interface ports

To configure 802.1x port authentication on a specific interface port, perform the following steps from privileged EXEC mode. Repeat this task for each interface port you wish to modify.

1. Enter the **configure** command to change to global configuration mode.

```
device# configure
```

2. Use the **interface** command to select the interface port to modify.

```
device(config)# interface tengigabitethernet 5/1/12
```

3. Use the **dot1x authentication** command to enable 802.1x authentication.

```
device(conf-if-te-5/1/12)# dot1x authentication
```

4. Return to privileged EXEC mode.

```
device(conf-if-te-5/1/12)# end
```

Configuring 802.1x timeouts on specific interface ports

NOTE

While you are free to modify the timeout values, Extreme recommends that you leave all timeouts set to their defaults.

To configure 802.1x timeout attributes on a specific interface port, perform the following steps from privileged EXEC mode. Repeat this task for each interface port you wish to modify.

1. Enter the **configure** command to change to global configuration mode.

```
device# configure
```

2. Use the **interface** command to select the interface port to modify.

```
device(config)# interface tengigabitethernet 5/1/12
```

3. Configure the **dot1x timeout** interval.

```
device(conf-if-te-5/1/12)# dot1x timeout supp-timeout 40
```

4. Return to privileged EXEC mode.

```
device(conf-if-te-5/1/12)# end
```

Configuring 802.1x port reauthentication on specific interface ports

To configure 802.1x port reauthentication on a specific interface port, perform the following steps from privileged EXEC mode. Repeat this task for each interface port you want to modify.

1. Enter the **configure** command to change to global configuration mode.

```
device# configure
```

2. Use the **interface** command to select the interface port to modify.

```
device(config)# interface tengigabitethernet 5/1/12
```

- Use the **dot1x authentication** command to enable 802.1x authentication for the interface port.

```
device(conf-if-te-5/1/12)# dot1x authentication
```

- Configure reauthentication for the interface port.

```
device(conf-if-te-5/1/12)# dot1x reauthentication
device(conf-if-te-5/1/12)# dot1x timeout re-authperiod 4000
```

- Return to privileged EXEC mode.

```
device(conf-if-te-5/1/12)# end
```

- Save the *running-config* file to the *startup-config* file.

```
device# copy running-config startup-config
```

Configuring 802.1x port-control on specific interface ports

To configure 802.1x port-control on a specific interface port, perform the following steps from privileged EXEC mode. Repeat this task for each interface port you want to modify.

- Use the **configure** command to change to global configuration mode.

```
device# configure
```

- Use the **interface** command to select the interface port to modify.

```
device(config)# interface tengigabitethernet 5/1/12
```

- Use the **dot1x authentication** command to enable 802.1x authentication for the interface port.

```
device(conf-if-te-5/1/12)# dot1x authentication
```

- Change the port authentication mode to **auto**, **force-authorized** or **force-unauthorized**.

```
device(conf-if-te-5/1/12)# dot1x port-control overlay auto
```

- Return to privileged EXEC mode.

```
device(conf-if-te-5/1/12)# end
```

Reauthenticating specific interface ports

To reauthenticate a supplicant connected to a specific interface port, perform the following steps from privileged EXEC mode. Repeat this task for each interface port you wish to reauthenticate.

- Use the **configure** command to change to global configuration mode.

```
device# configure
```

- Use the **interface** command to select the interface port to modify.

```
device(config)# interface tengigabitethernet 5/1/12
```

- Use the **dot1x reauthenticate** command to re-authenticate a port where dot1x is already enabled.

```
device(conf-if-te-5/1/12)# dot1x reauthenticate
```

- Return to privileged EXEC mode.

```
device(conf-if-te-5/1/12)# end
```

Disabling 802.1x on specific interface ports

To disable 802.1x authentication on a specific interface port, perform the following steps from privileged EXEC mode.

- Enter the **configure** command to change to global configuration mode.

```
device# configure
```

- Use the **interface** command to select the interface port to modify.

```
device(config)# interface tengigabitethernet 5/1/12
```

- Use the **no dot1x port-control** command to disable 802.1x authentication.

```
device(conf-if-te-5/1/12)# no dot1x port-control
```

- Return to privileged EXEC mode.

```
device(conf-if-te-5/1/12)# end
```

Disabling 802.1x globally

To disable 802.1x authentication globally, perform the following steps from privileged EXEC mode.

- Enter the **configure** command to change to global configuration mode.

```
device# configure
```

- Use the **no dot1x enable** command to disable 802.1x authentication.

```
device(config)# no dot1x enable
```

- Return to privileged EXEC mode.

```
device(config)# end
```

Checking 802.1x configurations

To check 802.1x configurations, perform the following steps from privileged EXEC mode.

- To view all dot1x configuration information, use the **show dot1x** command with the **all** keyword.

```
device# show dot1x all
```

- To check 802.1x configurations for specific interface ports, use the **interface** command to select the interface port to modify.

```
device(config)# interface tengigabitethernet 5/1/12
```

- To check 802.1x authentication statistics on specific interface ports, use the **show dot1x** command with the **statistics interface** keyword.

```
device# show dot1x statistics interface tengigabitethernet 5/1/12
```

4. To check all diagnostics information of the authenticator associated with a specific interface port, use the **show dot1x** command with the **diagnostics interface** keyword.

```
device# show dot1x diagnostics interface tengigabitethernet 5/1/12
```

5. To check all statistical information of the established session, use the **show dot1x** command with the **session-info interface** keyword.

```
device# show dot1x session-info interface tengigabitethernet 5/1/12
```

MAC authentication

In a network, many types of clients may gain access through publicly accessible ports and use the network resources. Such networks cannot be left unrestricted due to security concerns. There must be a mechanism to enforce authentication of the clients before allowing access to the network.

MAC authentication is a mechanism by which incoming traffic originating from a specific MAC address is switched or forwarded by the device only if the source MAC address is successfully authenticated by an authentication server. The RADIUS server is configured with a database of client MAC addresses that are allowed to gain access to the network. The MAC address itself is used as the username and password for authentication; and the user does not need to provide a specific username and password to gain network access.

When the authenticator device receives an Ethernet packet from the client, a RADIUS Access-Request packet containing the MAC address of the client is sent to the authentication server (RADIUS server). The RADIUS server looks up the database to validate the MAC address. If the MAC address is found in the database, the RADIUS server sends an Access-Accept packet to the device authenticating the client, and traffic from the client (MAC address) is forwarded normally.

The device supports multiple RADIUS servers; if communication with one of the RADIUS servers times out, the others are tried in sequential order. If a response from a RADIUS server is not received within a specified time (by default, 5 seconds) the RADIUS session times out, and the device retries the request up to 5 times. If no response is received, the next RADIUS server is chosen, and the request is sent for authentication.

If authorization fails, re-authentication attempt for the clients that are denied access will occur after a hold-off time of 300 seconds which is not configurable.

MAC authentication bypass

A single authentication method such as 802.1X authentication may not be compatible for all the clients that support different authentication methods.

Some clients such as printers and fax machines cannot respond to EAPOL messages to initiate 802.1X authentication. In such cases, it is not feasible to assign separate ports with specific authentication methods for different types of clients. MAC authentication bypass (MAB) provides a means to authenticate the client based on the MAC address, if the 802.1X authentication times out while waiting for an EAPOL message exchange.

MAB can be enabled only on an 802.1X authentication-enabled port. When the authenticator does not receive EAPOL response from the client to initiate 802.1X authentication after the maximum number of reauthentication attempts specified using the **dot1x reauthMax** command, MAB is triggered.

NOTE

The **dot1x reauthMax** command must be configured to a value from 3 through 10 to initiate MAB. If EAPOL response is received within the specified number of attempts, 802.1X authentication remains active and authentication request is sent to the RADIUS server.

When the port falls back to MAB mode, the device uses the MAC address as the client identity for authentication, and thereupon the port will not revert to the 802.1X authentication mode.

Dynamic VLAN assignment in MAC authentication and MAB

After successful MAC authentication, a VLAN assignment policy can be applied to control the destination of the client.

When a client or supplicant successfully completes the EAP authentication process, the authentication server (RADIUS server) sends the authenticator (the device) a RADIUS Access-Accept message that grants the client access to the network. Dynamic VLAN assignment allows clients to connect to the network anywhere and, based on their credentials, they get placed in the RADIUS-assigned VLAN irrespective of the ports to which they are connected.

For every MAC address, a VLAN can be assigned in the RADIUS server. The RADIUS Access-Accept message contains attributes set for the user in the user's access profile on the RADIUS server. A client is dynamically assigned to a VLAN based on the attribute sent from the RADIUS server. If one of the attributes in the Access-Accept message specifies a VLAN identifier (ID), and this VLAN is available on the device, the client's port is moved from its default VLAN to the specified VLAN. However, based on the VLAN response, device accepts only the first VLAN learned from the RADIUS server and all the subsequent authenticated clients are added to the same VLAN. The subsequent clients authenticated with different VLANs are rejected. When all MAC addresses learnt with dynamic VLAN are aged out, the interface is placed back in the default VLAN.

TABLE 13 RADIUS attributes for dynamic VLAN assignment

Attribute name	Type	Value
Tunnel-Type	064	13 (decimal) - VLAN
Tunnel-Medium-Type	065	6 (decimal) - 802
Tunnel-Private-Group-ID	081	<i>vlan-number</i> (decimal).

The device reads the attributes as follows:

- All three VLAN ID attributes (Tunnel-Private-Group-ID, Tunnel-Type, and Tunnel-Medium-Type) must be present in the response from the RADIUS server for VLAN processing.
- If the Tunnel-Type or Tunnel-Medium-Type attributes (or both) are not present, then the client is moved to the unauthorized state displaying an error message on the device.
- If the Tunnel-Type or Tunnel-Medium-Type attributes in the Access-Accept message have the values specified in the table, but there is no value specified for the Tunnel-Private-Group-ID attribute, the client will not become authorized.
- When the device receives the value specified for the Tunnel-Private-Group-ID attribute, it checks whether the *vlan-ID* matches the VLAN configured on the device. If there is a VLAN on the device that matches the *vlan-ID*, then the client's port is placed in the VLAN that corresponds to the VLAN ID.

If the RADIUS server does not assign any VLAN, the authenticated clients are added to the default VLAN. If the access port is configured with non-default VLAN, the client is moved to the non-default access VLAN irrespective of the VLAN assignment done in the RADIUS server. If MAC-based VLAN classifier is configured on the access port, the client is moved to the corresponding VLANs as per the VLAN classifier irrespective of the VLAN assignment done in the RADIUS server. The dynamically assigned VLAN will be removed and the device reverts to the default VLAN when the last authenticated MAC address is removed or ages out.

NOTE

VLAN assignment per supplicant is not supported.

Configuration notes for MAC authentication and MAB

- MAC authentication and MAB are supported only on Layer 2 switch ports configured as access ports. These authentication methods are not supported on port-channel members, port-channels, profiled port, ISL port, and trunk port.
- If 802.1X or MAB is enabled for a port, MAC authentication cannot be enabled on the same port.
- If the port authentication mode is set to force-authorized or force-unauthorized, the port does not fall back to MAB. That is, the port remains in port-based authentication mode itself.
- A maximum of 3000 clients (source MAC addresses) are supported for both MAC authentication and MAB respectively. This number includes both authenticated and non-authenticated MACs.
- MAC authentication and MAB are interoperable with port MAC security. If port MAC security is enabled on the port and the MAC limit is set to a value less than 3000 (MAC authentication scale limit) using the **switchport port-security max** command, the MAC limit set for port MAC security is honored for MAB and MAC authentication.
- If port MAC security is enabled on the same port, authentication request is sent only for the dynamically learnt MACs. Port MAC security secure MACs and sticky MACs are not supported.
- If ACL is applied on the access port or access VLAN, authentication request is sent only for the permitted packets and the MAC addresses are authenticated or denied as per the RADIUS database.
- MAC authentication is not supported with OUI security.
- All MAC authentication related configurations are persistent across the reboot.

Configuring MAC authentication bypass

To enable and activate MAC authentication bypass (MAB), perform the following steps.

MAB can be enabled only on an 802.1X authentication-enabled port and requires the same prerequisite tasks as for the 802.1X authentication. Before configuring MAB, communication between the devices and the authentication server must be established. The following configurations must be completed before configuring MAB:

- Configure the RADIUS server to authenticate access to the device. The **radius-server** command adds the RADIUS server to the device as the authentication server. This command can be repeated for additional servers. The **radius-server** command attempts to connect to the first RADIUS server. If the RADIUS server is not reachable, the next RADIUS server is contacted. If the RADIUS server is contacted and the authentication fails, the authentication process does not check for the next server in the sequence.

```
device(config)# radius-server host 10.0.0.5
```

1. (Optional) Enable the 802.1X readiness check on the device to determine if the devices connected to the ports are 802.1X-capable.

```
dot1x test eapol-capable interface Tengigabitethernet 5/1/12
DOT1X_PORT_EAPOL_CAPABLE:DOT1X: MAC 00-01-02-4b-f1-a3 on tengigabitethernet 5/1/12 is EAPOL capable.
```

2. Enter the **configure terminal** command to enter global configuration mode.

```
device# configure terminal
```

3. Enable 802.1X authentication globally.

```
device(config)# dot1x enable
```

If you globally disable 802.1X authentication, then all interface ports with 802.1X authentication enabled, automatically switch to force-authorized port control mode.

4. Enter interface configuration mode to configure interface-specific administrative features for 802.1X authentication.

```
device(config)# interface Tengigabitethernet 5/1/12
```

5. Define the interface in Layer 2 mode to set the switching characteristics of the Layer 2 interface.

```
device(conf-if-te-5/1/12)# switchport
```

All Layer 2 interfaces are mapped to default VLAN 1 and the interface is set to access mode. For changing the interface configuration mode to trunk or changing the default VLAN mapping, use additional switchport commands.

6. Enable 802.1X authentication on a specific interface port.

```
device(conf-if-te-5/1/12)# dot1x authentication
```

7. (Optional) Enter the dot1x port-control auto command to set the controlled port in the unauthorized state until authentication takes place between the client and the authentication server.

```
device(conf-if-te-5/1/12)# dot1x port-control auto
```

The port control is set to auto by default. The action activates authentication on an 802.1X-enabled interface. Once the client passes authentication, the port becomes authorized for that client. The controlled port remains in the authorized state for that client until the client logs off.

8. Configure the device to periodically reauthenticate the clients connected to 802.1X-enabled interfaces at regular intervals.

```
device(conf-if-te-5/1/12)# dot1x reauthentication
```

When you enable periodic reauthentication, the device reauthenticates the clients every 3,600 seconds by default.

9. (Optional) Configure the timeout parameters that determine the time interval for client reauthentication and EAP retransmissions using the following commands:

- Enter the **dot1x timeout re-authperiod** command to change and specify a different reauthentication interval.

```
device(conf-if-te-5/1/12)# dot1x timeout re-authperiod 300
```

- Enter the **dot1x timeout tx-period** command to change the amount of time the device should wait before retransmitting EAP-Request/Identity frames to the client.

```
device(conf-if-te-5/1/12)# dot1x timeout tx-period 30
```

- Enter the **dot1x timeout supp-timeout** command to change the amount of time the device should wait before retransmitting RADIUS EAP-Request/Challenge frames to the client.

```
device(conf-if-te-5/1/12)# dot1x timeout supp-timeout 30
```

Based on the timeout parameters, client reauthentication and retransmission of EAP-Request/Identity frames and EAP-Request/Challenge frames is performed.

10. Configure the maximum number of reauthentication attempts before the port goes to the unauthorized state.

```
device(conf-if-te-5/1/12)# dot1x reauthMax 3
```

The maximum number of reauthentication attempts must be configured to a value from 3 through 10 to initiate MAB. If EAPOL response is received within the specified number of attempts, 802.1X authentication remains active and authentication request is sent to the RADIUS server.

11. Configure MAC authentication bypass to authenticate the client based on the MAC address if the 802.1X authentication times out while waiting for an EAPOL message exchange.

```
device(conf-if-te-5/1/12)# dot1x mac-auth-bypass
```

Configuring MAC authentication

To enable and activate MAC authentication, perform the following steps.

MAC authentication requires some prerequisite tasks be performed before executing MAC authentication configurations at the interface level. Before configuring MAC authentication, communication between the devices and the authentication server must be established.

The following configurations must be completed before configuring MAC authentication:

- Configure the RADIUS server to authenticate access to the device. The **radius-server** command adds the RADIUS server to the device as the authentication server. This command can be repeated for additional servers. The **radius-server** command attempts to connect to the first RADIUS server. If the RADIUS server is not reachable, the next RADIUS server is contacted. If the RADIUS server is contacted and the authentication fails, the authentication process does not check for the next server in the sequence.

```
device(config)# radius-server host 10.0.0.5
```

1. (Optional) Enable the 802.1X readiness check on the device to determine if the devices connected to the ports are 802.1X-capable.

```
dot1x test eapol-capable interface tengigabitethernet 5/1/12
DOT1X_PORT_EAPOL_CAPABLE:DOT1X: MAC 00-01-02-4b-f1-a3 on tengigabitethernet 5/1/12 is EAPOL capable.
```

2. Enter the **configure terminal** command to enter global configuration mode.

```
device# configure terminal
```

3. Enable 802.1X authentication globally.

```
device(config)# dot1x enable
```

If you globally disable 802.1X authentication, then all interface ports with 802.1X authentication enabled, automatically switch to force-authorized port control mode.

4. Enter interface configuration mode to configure interface-specific administrative features for 802.1X authentication.

```
device(config)# interface Tengigabitethernet 5/1/12
```

5. Define the interface in Layer 2 mode to set the switching characteristics of the Layer 2 interface.

```
device(conf-if-te-5/1/12)# switchport
```

All Layer 2 interfaces are mapped to default VLAN 1 and the interface is set to access mode. For changing the interface configuration mode to trunk or changing the default VLAN mapping, use additional switchport commands.

6. Configure MAC authentication to authenticate the client based on the MAC address.

```
device(conf-if-te-5/1/12)# dot1x mac-auth-enable
```


Port MAC Security

- [Port MAC security overview.....](#)113
- [Configuring port MAC security.....](#)115

Port MAC security overview

Port MAC 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 MAC security features can be turned on to obtain the most robust port-security level that is appropriate. Basic port MAC security features are enabled in the device's default configuration. Additional features can be enabled with minimal configuration steps.

The following port MAC 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 MAC 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 MAC security configuration options

Port MAC security is disabled by default. The following table summarizes default port MAC security configuration options that are applied to an interface when it is made a secure port.

TABLE 14 Default configurations for port MAC security

Feature	Default configuration
Max. number of secure MAC addresses	8192
Violation mode	Shutdown
Shutdown time (minutes)	0

Port MAC security commands

Port MAC security is enabled on an interface by means of a series of switchport commands. For configuration examples, refer to [Configuring port MAC security](#) section and the *Network OS Command Reference*.

Command	Description
<code>switchport port-security</code>	Enables or disables port MAC security on an interface port.

Command	Description
switchport port-security mac-address	Configures the MAC address option for port MAC security on an interface port.
switchport port-security max	Configures the maximum number of MAC addresses used for port MAC security on an interface port.
switchport port-security oui	Configures an Organizationally Unique Identifier (OUI) MAC address for port MAC security on an interface port.
switchport port-security shutdown-time	Configures the shutdown-time option for port MAC 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 MAC security on an interface.

Port MAC 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 MAC security feature is applied.
show port-security	Displays the configuration information related to port MAC security.
show port-security addresses	Displays the configuration information related to port MAC security addresses.
show port-security interface	Displays the configuration information related to port MAC security interfaces.
show port-security oui interface	Displays the configuration information related to port MAC security for Organizationally Unique Identifier (OUI) interfaces.
show port-security sticky interface	Displays the configuration information related to port MAC security for a sticky interface

Port MAC security guidelines and restrictions

Note the following guidelines and restrictions for configuring port MAC security:

- A port mode change is not allowed when port MAC 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 MAC 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 MAC 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 MAC 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 MAC 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 MAC security feature.

Configuring port MAC security

The following section covers how to configure port MAC security for access and trunk ports, set port MAC security MAC address limits and shutdown time, set up OUI-based port security, and configure port MAC security with sticky MAC addresses.

Refer also to [Port MAC security overview](#) on page 113.

Configuring port MAC security on an access port

To enable port MAC security on an access port, do the following in global configuration mode.

1. Enable interface subconfiguration mode for the interface you want to modify.

```
device(config)# interface TenGigabitEthernet 1/0/1
```

2. Put the interface in Layer 2 mode by using the **switchport** command.

```
device(conf-if-te-1/0/1)# switchport
```

3. Enable switchport security by using the **switchport port-security** command.

```
device(conf-if-te-1/0/1)# switchport port-security
```

Configuring port MAC security on a trunk port

To enable port MAC security on a trunk port, do the following in global configuration mode.

1. Enable interface subconfiguration mode for the interface you want to modify.

```
device(config)# interface TenGigabitEthernet 1/0/1
```

2. Put the interface in Layer 2 mode by using the **switchport** command.

```
device(conf-if-te-1/0/1)# switchport
```

3. Set the mode of the interface to trunk.

```
device(conf-if-te-1/0/1)# switchport mode trunk
```

4. Set the VLANs that will transmit and receive through the Layer 2 interface.

```
device(conf-if-te-1/0/1)# switchport trunk allowed vlan add 100
```

5. Enable switch port security by using the **switchport port-security** command.

```
device(conf-if-te-1/0/1)# switchport port-security
```

Configuring port MAC security MAC address limits

To configure the MAC address option for port MAC security on an interface port, do the following in global configuration mode.

1. Enable interface subconfiguration mode for the interface you want to modify.

```
device(config)# interface TenGigabitEthernet 1/0/1
```

- Put the interface in Layer 2 mode by using the **switchport** command.

```
device(conf-if-te-1/0/1)# switchport
```

- Set the MAC address and VLAN ID for the interface.

```
device(conf-if-te-1/0/1)# switchport port-security mac-address 1000.2000.3000 vlan 100
```

Configuring port MAC 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 MAC security shutdown time for an interface port, do the following in global configuration mode.

- Enable interface subconfiguration mode for the interface you want to modify.

```
device(config)# interface TenGigabitEthernet 1/0/1
```

- Put the interface in Layer 2 mode by using the **switchport** command.

```
device(conf-if-te-1/0/1)# switchport
```

- Set the violation response option to shutdown.

```
device(conf-if-te-1/0/1)# switchport port-security violation shutdown
```

- Set the shutdown time, in minutes.

```
device(conf-if-te-1/0/1)# switchport port-security shutdown-time 10
```

Configuring OUI-based port MAC 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 MAC security, do the following in global configuration mode.

- Enable interface subconfiguration mode for the interface you want to modify.

```
device(config)# interface TenGigabitEthernet 1/0/1
```

- Put the interface in Layer 2 mode by using the **switchport** command.

```
device(conf-if-te-1/0/1)# switchport
```

- Configure a permitted OUI MAC address by using the **switchport port-security oui** command.

```
device(conf-if-te-1/0/1)# switchport port-security oui 2000.3000.4000
```

Configuring port MAC 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.

```
device(config)# interface TenGigabitEthernet 1/0/1
```

2. Put the interface in Layer 2 mode by using the **switchport** command.

```
device(conf-if-te-1/0/1)# switchport
```

3. Enable switchport security by using the **switchport port-security oui** command.

```
device(conf-if-te-1/0/1)# switchport port-security oui 2000.3000.4000
```

4. Configure the sticky option.

```
device(conf-if-te-1/0/1)# switchport port-security sticky
```


SSH - Secure Shell

- [Configuring SSH encryption protocol119](#)

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.
- Configure the maximum number of SSH sessions.

Ciphers, non-CBC ciphers, algorithms, and MACs are not mutually exclusive. Any combination of these items may 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. Shutdown and restart the SSH server using the **ssh server shutdown** command.

```
device(config)# ssh server shutdown  
device(config)# no ssh server shutdown
```

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

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. Shutdown and restart the SSH server using the **ssh server shutdown** command.

```
device(config)# ssh server shutdown
device(config)# no ssh server shutdown
```

6. Confirm the cipher setting with the **show running-config** command to set the client cipher version for SSH.

7. 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 **no 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 **no 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-hellman-group-14-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 configure terminal mode.

```
device#configure terminal
```

2. Enter RBridge ID mode.

```
device(config)#rbridge-id 3
```

3. Use the **no ssh server key-exchange** command to reset the key exchange algorithm for the server to the default value.
4. Use the **no ssh client key-exchange** command to reset the key exchange algorithm for the client to the default value.

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.

```
device#configure terminal
```

2. Enter RBridge ID mode.

```
device(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.

```
device(config-rbridge-id-176)# ssh server mac hmac-sha1,hmac-sha2-256,hmac-sha2-512
```

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.

```
device(config-rbridge-id-176)# ssh client mac hmac-sha1,hmac-sha2-256,hmac-sha2-512
```

5. Restart the SSH server using the **no ssh server shutdown** command.
6. Enter the **show ssh** command to confirm the SSH configuration information.

```
device(config-rbridge-id-176)# do show ssh server status
rbridge-id 176:SSH Server Mac : hmac-md5,hmac-sha1,hmac-sha2-256 rbridge-id 176
VRF-Name: mgmt-vrf      Status: Enabled
VRF-Name: default-vrf   Status: Enabled
```

Removing an SSH MAC

Removes SSH Server and Client Message Authentication Codes (MACs).

The "no" form of the **ssh server mac** and **ssh client mac** commands removes the MACs.

1. Enter configure terminal mode.

```
device# configure terminal
```

2. Enter RBridge ID mode.

```
device(config)#rbridge-id 176
```

3. On the SSH server, enter the **no ssh server mac** command to set the SSH server MACs to default values.
4. Restart the SSH server using the **no ssh server shutdown** command.
5. On the SSH client, enter the **no ssh client mac** command to set the SSH server MACs to default values.

Managing SSH public keys

You can import SSH public keys to establish an authenticated login for a device. You can also delete the key from the device to prevent it from being used for an authenticated login.

To manage the SSH keys, perform the following steps:

1. In privileged EXEC mode, import an SSH public key to the device.

```
device# certutil import sshkey user admin host 10.70.4.106 directory /users/home40/bmeenaks/.ssh
file id_rsa.pub login fvt
```

This example imports the SSH public key for the admin user from the remote 10.70.4.106 host using the directory and file information to the key.

2. Enter the password for the user.

```
Password: *****
```

When the SSH key is imported, the following message appears.

```
device# 2016/01/14-10:28:58, [SEC-3050], 75,, INFO, VDX6740-48, Event: sshutil, Status: success,
Info: Imported SSH public key from 10.70.4.106 for user 'admin'.
```

3. Delete an SSH public key from the device prevents it from being used.

```
device# no certutil sshkey user admin
```

This example deletes the SSH key for the admin user.

Importing an SSH public key

Importing an SSH public key allows you to establish an authenticated login for a switch.

You must be in privileged EXEC mode to import an SSH public key to a switch.

1. To import an SSH public key, enter **certutil import sshkey**, followed by **user Username host IP_Address directory File_Path file Key_filename login Login_ID**.

```
device# certutil import sshkey user admin host 10.70.4.106 directory /users/home40/bmeenaks/.ssh
file id_rsa.pub login fvt
```

This enables you to import the SSH public key for the user "admin" from a remote host.

2. Enter the password for the user.

```
Password: *****
```

```
device# 2012/11/14-10:28:58, [SEC-3050], 75,, INFO, VDX6740-48, Event: sshutil, Status: success,
Info: Imported SSH public key from 10.70.4.106 for user 'admin'.
```

NOTE

In a VCS Fabric, you must enter RBridge ID configuration mode before issuing the command.

```
device# certutil import sshkey user admin host 10.70.4.106 directory /users/home40/bmeenaks/.ssh file
id_rsa.pub login fvt rbridge-id 3
```

Deleting an SSH public key

Deleting an SSH public key from a switch prevents it from being used for an authenticated login.

You must be in privileged EXEC mode to delete an SSH public key from a switch.

To delete an SSH public key, enter **no certutil sshkey user Username** followed by either **rbridge-id rbridge-id** or **rbridge-id all**.

```
device# no certutil sshkey user admin rbridge-id all
```

Specifying a specific RBridge ID removes the key from that RBridge ID; specifying all removes it from all RBridge IDs on the switch.

Configuring self-signed certificates for VXLAN gateways

For the details of generating and removing a self-signed certificate on a VXLAN gateway, refer to "VXLAN gateway and NSX Controller deployments" in the *Network OS Layer 2 Switching Configuration Guide*.

Removing self-signed certificates for VXLAN gateways

Use the **nsx-controller client-cert delete** command to remove the certificate.

1. Use the **nsx-controller client-cert delete** command to remove the certificate.
You can remove multiple ciphers by separating the string names with commas.

```
device# nsx-controller client-cert delete
```

2. Use the **show nsx-controller client-cert** command to verify the configuration.

Configuring the maximum number of SSH sessions

An SSH server can reuse an already established TCP connection for multiple sessions (also known as multiplexing). This reduces the time required to open a new connection for each SSH session, as well as the overhead of allocating separate resources for each connection.

SSH clients must also be configured to support multiplexing, in accordance with local best practices.

Note the following additional usage guidelines.

After executing this command, in order to use the new number of sessions, you must first shut down the SSH server, by means of the **ssh server use-vrf shutdown** command, and then restart it, by means of the **no ssh server use-vrf shutdown** command.

The maximum number of sessions specified by this command is synchronized to the standby management module (MM). However, to make the change effective on the standby MM, you must first disable service on that module by means of the **no ssh server standby enable** command, and then reenables service by means of the **ssh server standby enable** command.

Use the **show running-config rbridge-id ssh server** command or the **show ssh server status** command to confirm the configuration.

A downgrade to a previous release is blocked if this command has been executed in the running configuration.

Use the **no ssh server max-sessions** command to revert to the default of 1 session. You must also stop and restart service as in the usage guidelines above.

1. From global configuration mode, enter RBridge ID configuration mode for a specified RBridge.

```
device# configure terminal
device(config)# rbridge-id 176
device(config-rbridge-id-176)#
```

2. Enter the **ssh server max-sessions** command and specify the maximum number of sessions to be supported. (Range is from 1 through 10.)
3. Use the **show running-config rbridge-id ssh server** command in this mode to confirm the running configuration, which includes key types as well as the maximum number of SSH sessions configured.

```
device(config-rbridge-id-176)# do show running-config rbridge-id ssh server
rbridge-id 176
ssh server max-sessions 7
ssh server key rsa 2048
ssh server key ecdsa 256
ssh server key dsa
```

4. You can also use the **show running-config rbridge-id ssh server** command in this mode to view the maximum number of SSH sessions configured, as well as VRF status.

```
device(config-rbridge-id-176)# do show ssh server status rbridge-id 176
rbridge-id 176:
VRF-name: mgmt-vrf           Status: Enabled
VRF-name: default-vrf       Status: Enabled
rbridge-id 176: SSH Server Max sessions: 7
```


Router Advertisement (RA) Guard

- RA Guard overview..... 127
- RA Guard configuration guidelines 127
- Enabling and disabling RA Guard 128
- RA Guard Show commands..... 128

RA Guard overview

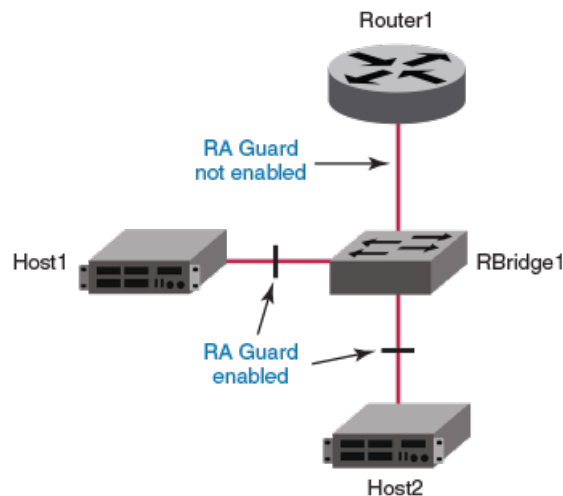
In a routed network, devices are configured to send router advertisements (RAs). RAs enable link nodes to discover routers, allowing the nodes to autoconfigure.

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 75.

You can apply RA Guard only on the Physical Switchport interface type.

RA Guard is only supported on the VDX 6740 and the 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.

```
device# configure
```

2. Enter the **interface** command, specifying the interface type and the rbridge-id/slot/port number.

```
device(config)# interface ten 122/5/22
```

3. To enable RA Guard on this interface, enter **ipv6 raguard**.

```
device(conf-if-te-122/5/22)# ipv6 raguard
```

4. To disable RA Guard on this interface, enter **no ipv6 raguard**.

```
device(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 15 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.