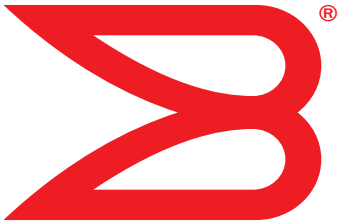
53-1002601-01
28 September 2012



Brocade ICX 6650

Security Configuration Guide

Supporting FastIron Software Release 07.5.00

BROCADE

Brocade Communications Systems, Incorporated

Corporate and Latin American Headquarters
Brocade Communications Systems, Inc.
130 Holger Way
San Jose, CA 95134
Tel: 1-408-333-8000
Fax: 1-408-333-8101
E-mail: info@brocade.com

Asia-Pacific Headquarters
Brocade Communications Systems China HK, Ltd.
No. 1 Guanghua Road
Chao Yang District
Units 2718 and 2818
Beijing 100020, China
Tel: +8610 6588 8888
Fax: +8610 6588 9999
E-mail: china-info@brocade.com

European Headquarters
Brocade Communications Switzerland Sàrl
Centre Swissair
Tour B - 4ème étage
29, Route de l'Aéroport
Case Postale 105
CH-1215 Genève 15
Switzerland
Tel: +41 22 799 5640
Fax: +41 22 799 5641
E-mail: emea-info@brocade.com

Asia-Pacific Headquarters
Brocade Communications Systems Co., Ltd. (Shenzhen WFOE)
Citic Plaza
No. 233 Tian He Road North
Unit 1308 – 13th Floor
Guangzhou, China
Tel: +8620 3891 2000
Fax: +8620 3891 2111
E-mail: china-info@brocade.com

Document History

Title	Publication number	Summary of changes	Date
Brocade ICX 6650 Security Configuration Guide	53-1002601-01	Release 07.4.00 document updated with enhancements in Release 07.5.00	September 2012

Contents

About This Document

The Brocade ICX 6650 is a ToR (Top of Rack) Ethernet switch for campus LAN and classic Ethernet data center environments.

Audience

This document is designed for system administrators with a working knowledge of Layer 2 and Layer 3 switching and routing.

If you are using a Brocade Layer 3 Switch, you should be familiar with the following protocols if applicable to your network: IP, RIP, OSPF, BGP, ISIS, PIM, and VRRP.

Supported hardware and software

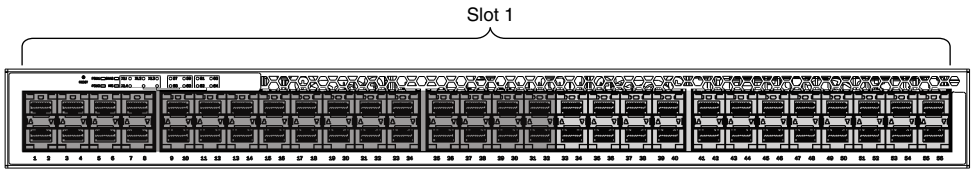
This document is specific to the Brocade ICX 6650 running FastIron 7.5.00.

Brocade ICX 6650 slot and port numbering

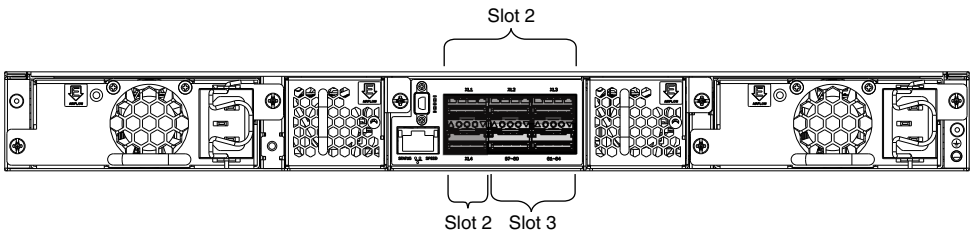
Many CLI commands require users to enter port numbers as part of the command syntax, and many **show** command outputs display port numbers. The port numbers are entered and displayed in stack-unit/slot number/port number format. In all Brocade ICX 6650 inputs and outputs, the stack-unit number is always 1.

The Brocade ICX 6650 contains the following slots and Ethernet ports:

- Slot 1 is located on the front of the Brocade ICX 6650 device and contains ports 1 through 56. Ports 1 through 32 are 10 GbE. Ports 33 through 56 are 1/10 GbE SFP+ ports. Refer to the following figure.



- Slot 2 is located on the back of the Brocade ICX 6650 device and contains ports 1 through 3 on the top row and port 4 on the bottom row. These ports are 2x40 GbE QSFP+. Refer to the following figure.



- Slot 3 is located on the back of the Brocade ICX 6650 device and contains ports 1 through 8. These ports are 4 x 10 GbE breakout ports and require the use of a breakout cable. Refer to the previous figure.

How this document is organized

This document is organized to help you find the information that you want as quickly and easily as possible.

The document contains the following components:

Document conventions

This section describes text formatting conventions and important notice formats used in this document.

Text formatting

The narrative-text formatting conventions that are used are as follows:

bold text	Identifies command names Identifies the names of user-manipulated GUI elements Identifies keywords and operands Identifies text to enter at the GUI or CLI
italic text	Provides emphasis Identifies variables Identifies paths and Internet addresses Identifies document titles
`code` text	Identifies CLI output Identifies command syntax examples

For readability, command names in the narrative portions of this guide are presented in mixed lettercase: for example, **switchShow**. In actual examples, command lettercase is all lowercase.

Command syntax conventions

Command syntax in this manual follows these conventions:

command	Commands are printed in bold.
--**option, option**	Command options are printed in bold.
-**argument,** arg	Arguments.
[ ]	Optional elements appear in brackets.
variable	Variables are printed in italics. In the help pages, values are underlined or enclosed in angled brackets < >.
...	Repeat the previous element, for example "member[;member...]"
value	Fixed values following arguments are printed in plain font. For example, --**show** WWN
|	Boolean. Elements are exclusive. Example: --**show** -**mode** egress | ingress

Notes, cautions, and warnings

The following notices and statements are used in this manual. They are listed below in 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 potential damage to hardware or data.

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.

Notice to the reader

This document might contain references to the trademarks of the following corporations. These trademarks are the properties of their respective companies and corporations.

These references are made for informational purposes only.

Corporation	Referenced Trademarks and Products
Microsoft Corporation	Windows, Windows NT, Internet Explorer
Oracle Corporation	Oracle, Java
Netscape Communications Corporation	Netscape
Mozilla Corporation	Mozilla Firefox
Sun Microsystems, Inc.	Sun, Solaris
Red Hat, Inc.	Red Hat, Red Hat Network, Maximum RPM, Linux Undercover

Related publications

The following Brocade documents supplement the information in this guide:

- *Brocade ICX 6650 Release Notes*
- *Brocade ICX 6650 Hardware Installation Guide New*
- *Brocade ICX 6650 Administration Guide*
- *Brocade ICX 6650 Platform and Layer 2 Configuration Guide*
- *Brocade ICX 6650 Layer 3 Routing Configuration Guide*
- *Brocade ICX 6650 Security Configuration Guide*
- *Brocade ICX 6650 IP Multicast Configuration Guide*

- *Brocade ICX 6650 Diagnostic Reference*
- *Unified IP MIB Reference*
- *Ports-on-Demand Licensing for the Brocade ICX 6650*

The latest versions of these guides are posted at *http://www.brocade.com/ethernetproducts*.

Additional information

This section lists additional Brocade and industry-specific documentation that you might find helpful.

Brocade resources

To get up-to-the-minute information, go to *http://my.brocade.com* to register at no cost for a user ID and password.

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

http://www.brocade.com/products-solutions/products/index.page

For additional Brocade documentation, visit the Brocade website:

http://www.brocade.com

Release notes are available on the MyBrocade website.

Other industry resources

For additional resource information, visit the Technical Committee T11 website. This website provides interface standards for high-performance and mass storage applications for Fibre Channel, storage management, and other applications:

http://www.t11.org

For information about the Fibre Channel industry, visit the Fibre Channel Industry Association website:

http://www.fibrechannel.org

Getting technical help

To contact Technical Support, go to

http://www.brocade.com/services-support/index.page

for the latest e-mail and telephone contact information.

Document feedback

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

documentation@brocade.com

Provide the title and version number of the document and as much detail as possible about your comment, including the topic heading and page number and your suggestions for improvement.

Chapter

1

Security Access

Table 1 lists the security access features supported on Brocade ICX 6650. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 1 Supported security access features

Feature	Brocade ICX 6650
Authentication, Authorization and Accounting (AAA): • RADIUS • TACACS/TACACS+	Yes
AAA support for console commands	Yes
Restricting remote access to management functions	Yes
Disabling TFTP access	Yes
Using ACLs to restrict remote access	Yes
Local user accounts	Yes
Local user passwords	Yes
AAA authentication-method lists	Yes
Packet filtering on TCP flags	Yes

This chapter explains how to secure access to management functions on a Brocade device.

NOTE
For the Brocade ICX 6650, RADIUS Challenge is supported for 802.1x authentication but not for login authentication. Also, multiple challenges are supported for TACACS+ login authentication.

Securing access methods

The following table lists the management access methods available on a Brocade device, how they are secured by default, and the ways in which they can be secured.

TABLE 2 Ways to secure management access to Brocade devices

Access method	How the access method is secured by default	Ways to secure the access method
Serial access to the CLI	Not secured	Establish passwords for management privilege levels

TABLE 2 Ways to secure management access to Brocade devices (Continued)

Access method	How the access method is secured by default	Ways to secure the access method
Access to the Privileged EXEC and CONFIG levels of the CLI	Not secured	Establish a password for Telnet access to the CLI
		Establish passwords for management privilege levels
		Set up local user accounts
		Configure TACACS/TACACS+ security
		Configure RADIUS security
Telnet access	Not secured	Regulate Telnet access using ACLs
		Allow Telnet access only from specific IP addresses
		Restrict Telnet access based on a client MAC address
		Allow Telnet access only from specific MAC addresses
		Define the Telnet idle time
		Change the Telnet login timeout period
		Specify the maximum number of login attempts for Telnet access
		Disable Telnet access
		Establish a password for Telnet access
		Establish passwords for privilege levels of the CLI
		Set up local user accounts
		Configure TACACS/TACACS+ security
		Configure RADIUS security
Secure Shell (SSH) access	Not configured	Configure SSH
		Regulate SSH access using ACLs
		Allow SSH access only from specific IP addresses
		Allow SSH access only from specific MAC addresses
		Establish passwords for privilege levels of the CLI
		Set up local user accounts
		Configure TACACS/TACACS+ security
		Configure RADIUS security

TABLE 2 Ways to secure management access to Brocade devices (Continued)

Access method	How the access method is secured by default	Ways to secure the access method
SNMP access	SNMP read or read-write community strings and the password to the Super User privilege level **NOTE:** SNMP read or read-write community strings are always required for SNMP access to the device.	Regulate SNMP access using ACLs
		Allow SNMP access only from specific IP addresses
		Disable SNMP access
		Allow SNMP access only to clients connected to a specific VLAN
		Establish passwords to management levels of the CLI
		Set up local user accounts
		Establish SNMP read or read-write community strings
TFTP access	Not secured	Allow TFTP access only to clients connected to a specific VLAN
		Disable TFTP access
Access for Stacked Devices	Access to multiple consoles must be secured after AAA is enabled	Extra steps must be taken to secure multiple consoles in an IronStack.

Remote access to management function restrictions

You can restrict access to management functions from remote sources, including Telnet and SNMP. The following methods for restricting remote access are supported:

- Using ACLs to restrict Telnet or SNMP access
- Allowing remote access only from specific IP addresses
- Allowing Telnet and SSH access only from specific MAC addresses
- Allowing remote access only to clients connected to a specific VLAN
- Specifically disabling Telnet or SNMP access to the device

The following sections describe how to restrict remote access to a Brocade device using these methods.

ACL usage to restrict remote access

You can use standard ACLs to control the following access methods to management functions on a Brocade device:

- Telnet
- SSH
- SNMP

Consider the following to configure access control for these management access methods.

1. Configure an ACL with the IP addresses you want to allow to access the device.
2. Configure a Telnet access group, SSH access group, and SNMP community strings. Each of these configuration items accepts an ACL as a parameter. The ACL contains entries that identify the IP addresses that can use the access method.

The following sections present examples of how to secure management access using ACLs. Refer to Chapter 3, "Rule-Based IP ACLs" for more information on configuring ACLs.

Using an ACL to restrict Telnet access

To configure an ACL that restricts Telnet access to the device, enter commands such as the following.

```
Brocade(config)# access-list 10 deny host 10.157.22.32 log
Brocade(config)# access-list 10 deny 10.157.23.0 0.0.0.255 log
Brocade(config)# access-list 10 deny 10.157.24.0 0.0.0.255 log
Brocade(config)# access-list 10 deny 10.157.25.0/24 log
Brocade(config)# access-list 10 permit any
Brocade(config)# telnet access-group 10
Brocade(config)# write memory
```

Syntax: telnet access-group *num*

The *num* parameter specifies the number of a standard ACL and must be from 1–99.

The commands above configure ACL 10, then apply the ACL as the access list for Telnet access. The device allows Telnet access to all IP addresses except those listed in ACL 10.

To configure a more restrictive ACL, create permit entries and omit the **permit any** entry at the end of the ACL.

Example

```
Brocade(config)# access-list 10 permit host 10.157.22.32
Brocade(config)# access-list 10 permit 10.157.23.0 0.0.0.255
Brocade(config)# access-list 10 permit 10.157.24.0 0.0.0.255
Brocade(config)# access-list 10 permit 10.157.25.0/24
Brocade(config)# telnet access-group 10
Brocade(config)# write memory
```

The ACL in this example permits Telnet access only to the IP addresses in the **permit** entries and denies Telnet access from all other IP addresses.

Using an ACL to restrict SSH access

To configure an ACL that restricts SSH access to the device, enter commands such as the following.

```
Brocade(config)# access-list 12 deny host 10.157.22.98 log
Brocade(config)# access-list 12 deny 10.157.23.0 0.0.0.255 log
Brocade(config)# access-list 12 deny 10.157.24.0/24 log
Brocade(config)# access-list 12 permit any
Brocade(config)# ssh access-group 12
Brocade(config)# write memory
```

Syntax: ssh access-group *num*

The *num* parameter specifies the number of a standard ACL and must be from 1–99.

These commands configure ACL 12, then apply the ACL as the access list for SSH access. The device denies SSH access from the IP addresses listed in ACL 12 and permits SSH access from all other IP addresses. Without the last ACL entry for permitting all packets, this ACL would deny SSH access from all IP addresses.

NOTE
In this example, the command **ssh access-group 10** could have been used to apply the ACL configured in the example for Telnet access. You can use the same ACL multiple times.

Using ACLs to restrict SNMP access

To restrict SNMP access to the device using ACLs, enter commands such as the following.

NOTE
The syntax for using ACLs for SNMP access is different from the syntax for controlling Telnet, SSH, access using ACLs.

```
Brocade(config)# access-list 25 deny host 10.157.22.98 log
Brocade(config)# access-list 25 deny 10.157.23.0 0.0.0.255 log
Brocade(config)# access-list 25 deny 10.157.24.0 0.0.0.255 log
Brocade(config)# access-list 25 permit any
Brocade(config)# access-list 30 deny 10.157.25.0 0.0.0.255 log
Brocade(config)# access-list 30 deny 10.157.26.0/24 log
Brocade(config)# access-list 30 permit any
Brocade(config)# snmp-server community public ro 25
Brocade(config)# snmp-server community private rw 30
Brocade(config)# write memory
```

Syntax: snmp-server community *string* **ro** | **rw** *num*

The *string* parameter specifies the SNMP community string you must enter to gain SNMP access.

The **ro** parameter indicates that the community string is for read-only (“get”) access. The **rw** parameter indicates the community string is for read-write (“set”) access.

The *num* parameter specifies the number of a standard ACL and must be from 1 – 99.

These commands configure ACLs 25 and 30, then apply the ACLs to community strings.

ACL 25 is used to control read-only access using the “public” community string. ACL 30 is used to control read-write access using the “private” community string.

NOTE
When **snmp-server community** is configured, all incoming SNMP packets are validated first by their community strings and then by their bound ACLs.

Defining the console idle time

By default, a Brocade device does not time out serial console sessions. A serial session remains open indefinitely until you close it. You can however define how many minutes a serial management session can remain idle before it is timed out.

NOTE
You must enable AAA support for console commands, AAA authentication, and Exec authorization in order to set the console idle time.

To configure the idle time for a serial console session, use the following command.

```
Brocade(config)# console timeout 120
```

Syntax: [no] **console timeout** *minutes*

Possible values for the *minutes* variable: 0–240 minutes

Default value: 0 minutes (no timeout)

NOTE
In RADIUS, the standard attribute Idle-Timeout is used to define the console session timeout value. The attribute Idle-Timeout value is specified in seconds. Within the switch, it is truncated to the nearest minute, because the switch configuration is defined in minutes.

Remote access restrictions

By default, a Brocade device does not control remote management access based on the IP address of the managing device. You can restrict remote management access to a single IP address for the following access methods:

- Telnet access
- SSH access
- SNMP access

In addition, you can restrict all access methods to the same IP address using a single command.

The following examples show the CLI commands for restricting remote access. You can specify only one IP address with each command. However, you can enter each command ten times to specify up to ten IP addresses.

Restricting Telnet access to a specific IP address

To allow Telnet access to the Brocade device only to the host with IP address 10.157.22.39, enter the following command.

```
Brocade(config)# telnet-client 10.157.22.39
```

Syntax: [no] **telnet-client** *ip-addr* | *ipv6-addr*

Restricting SSH access to a specific IP address

To allow SSH access to the Brocade device only to the host with IP address 10.157.22.39, enter the following command.

```
Brocade(config)# ip ssh client 10.157.22.39
```

Syntax: [no] **ip ssh client** *ip-addr* | *ipv6-addr*

Restricting SNMP access to a specific IP address

To allow SNMP access only to the host with IP address 10.157.22.14, enter the following command.

```
Brocade(config)# snmp-client 10.157.22.14
```

Syntax: [**no**] **snmp-client** *ip-addr* | *ipv6-addr*

Restricting all remote management access to a specific IP address

To allow Telnet and SNMP management access to the Brocade device only to the host with IP address 10.157.22.69, enter three separate commands (one for each access type) or enter the following command.

```
Brocade(config)# all-client 10.157.22.69
```

Syntax: [**no**] **all-client** *ip-addr* | *ipv6-addr*

Restricting access to the device based on IP or MAC address

You can restrict remote management access to the Brocade device, using Telnet, SSH, HTTP, and HTTPS, based on the connecting client IP or MAC address.

Restricting Telnet connection

You can restrict Telnet connection to a device based on the client IP address or MAC address.

To allow Telnet access to the Brocade device only to the host with IP address 10.157.22.39 and MAC address 0000.000f.e9a0, enter the following command.

```
Brocade(config)# telnet client 10.157.22.39 0000.000f.e9a0
```

Syntax: [**no**] **telnet client** *ip-addr* | *ipv6-addr mac-addr*

The following command allows Telnet access to the Brocade device to a host with any IP address and MAC address 0000.000f.e9a0.

```
Brocade(config)# telnet client any 0000.000f.e9a0
```

Syntax: [**no**] **telnet client any** *mac-addr*

Restricting SSH connection

You can restrict SSH connection to a device based on the client IP address or MAC address.

To allow SSH access to the Brocade device only to the host with IP address 10.157.22.39 ***and*** MAC address 0000.000f.e9a0, enter the following command.

```
Brocade(config)# ip ssh client 10.157.22.39 0000.000f.e9a0
```

Syntax: [**no**] **ip ssh client** *ip-addr* | *ipv6-addr mac-addr*

To allow SSH access to the Brocade device to a host with any IP address and MAC address 0000.000f.e9a0, enter the following command.

```
Brocade(config)# ip ssh client any 0000.000f.e9a0
```

Syntax: [no] **ip ssh client any** *mac-addr*

Restricting HTTP and HTTPS connection

You can restrict an HTTP or HTTPS connection to a device based on the client IP address or MAC address.

To allow HTTP and HTTPS access to the Brocade device only to the host with IP address 10.157.22.40 and MAC address 0000.000f.ab1c, enter the following command.

```
Brocade(config)# web client 10.157.22.40 0000.000f.ab1c
```

Syntax: [no] **web client** *ip-addr* | *ipv6-addr mac-addr*

The following command allows HTTP and HTTPS access to the Brocade device to a host with any IP address and MAC address 0000.000f.10ba.

```
Brocade(config)# web client any 0000.000f.10ba
```

Syntax: [no] **web client any** *mac-addr*

Defining the Telnet idle time

You can define how many minutes a Telnet session can remain idle before it is timed out. An idle Telnet session is a session that is still sending TCP ACKs in response to keepalive messages from the device, but is not being used to send data.

To configure the idle time for a Telnet session, use the following command.

```
Brocade(config)# telnet timeout 120
```

Syntax: [no] **telnet timeout** *minutes*

For *minutes* enter a value from 0–240. The default value is 0 minutes (no timeout).

Changing the login timeout period for Telnet sessions

By default, the login timeout period for a Telnet session is 1 minute. To change the login timeout period, use the following command.

```
Brocade(config)# telnet login-timeout 5
```

Syntax: [no] **telnet login-timeout** *minutes*

For *minutes*, enter a value from 1 to 10. The default timeout period is 1 minute.

Specifying the maximum number of login attempts for Telnet access

If you are connecting to the Brocade device using Telnet, the device prompts you for a username and password. By default, you have up to 4 chances to enter a correct username and password. If you do not enter a correct username or password after 4 attempts, the Brocade device disconnects the Telnet session.

You can specify the number of attempts a Telnet user has to enter a correct username and password before the device disconnects the Telnet session. For example, to allow a Telnet user up to 5 chances to enter a correct username and password, enter the following command.

```
Brocade(config)# telnet login-retries 5
```

Syntax: [**no**] **telnet login-retries** *number*

You can specify from 0–5 attempts. The default is 4 attempts.

Changing the login timeout period for Telnet sessions

To change the login timeout period for Telnet sessions to 5 minutes, enter the following command:

```
Brocade(config)# telnet login-timeout 5
```

Syntax: [**no**] **telnet login-timeout** *minutes*

For *minutes*, specify a value from 1–10. The default is 2 minutes.

Restricting remote access to the device to specific VLAN IDs

You can restrict management access to a Brocade device to ports within a specific port-based VLAN. VLAN-based access control applies to the following access methods:

- Telnet access
- SNMP access
- TFTP access

By default, access is allowed for all the methods listed above on all ports. After you configure security for a given access method based on VLAN ID, access to the device using that method is restricted to only the ports within the specified VLAN.

VLAN-based access control works in conjunction with other access control methods. For example, suppose you configure an ACL to permit Telnet access only to specific client IP addresses, and you also configure VLAN-based access control for Telnet access. In this case, the only Telnet clients that can access the device are clients that have one of the IP addresses permitted by the ACL ***and*** are connected to a port that is in a permitted VLAN. Clients who have a permitted IP address but are connected to a port in a VLAN that is not permitted still cannot access the device through Telnet.

Restricting Telnet access to a specific VLAN

To allow Telnet access only to clients in a specific VLAN, enter a command such as the following.

```
Brocade(config)# telnet server enable vlan 10
```

The command in this example configures the device to allow Telnet management access only to clients connected to ports within port-based VLAN 10. Clients connected to ports that are not in VLAN 10 are denied management access.

Syntax: [no] **telnet server enable vlan** *vlan-id*

Restricting SNMP access to a specific VLAN

To allow SNMP access only to clients in a specific VLAN, enter a command such as the following.

```
Brocade(config)# snmp-server enable vlan 40
```

The command in this example configures the device to allow SNMP access only to clients connected to ports within port-based VLAN 40. Clients connected to ports that are not in VLAN 40 are denied access.

Syntax: [no] **snmp-server enable vlan** *vlan-id*

Restricting TFTP access to a specific VLAN

To allow TFTP access only to clients in a specific VLAN, enter a command such as the following.

```
Brocade(config)# tftp client enable vlan 40
```

The command in this example configures the device to allow TFTP access only to clients connected to ports within port-based VLAN 40. Clients connected to ports that are not in VLAN 40 are denied access.

Syntax: [no] **tftp client enable vlan** *vlan-id*

Designated VLAN for Telnet management sessions to a Layer 2 switch

Brocade ICX 6650 supports the creation of management VLANs. By default, the management IP address you configure on a Layer 2 switch applies globally to all the ports on the device. This is true even if you divide the device ports into multiple port-based VLANs.

If you want to restrict the IP management address to a specific port-based VLAN, you can make that VLAN the designated management VLAN for the device. When you configure a VLAN to be the designated management VLAN, the management IP address you configure on the device is associated only with the ports in the designated VLAN. To establish a Telnet management session with the device, you must access the device through one of the ports in the designated VLAN.

You also can configure up to five default gateways for the designated VLAN, and associate a metric with each one. The software uses the gateway with the lowest metric. The other gateways reside in the configuration but are not used. To use one of the other gateways, modify the configuration so that the gateway you want to use has the lowest metric.

If more than one gateway has the lowest metric, the gateway that appears first in the running-config is used.

NOTE
If you have already configured a default gateway globally and you do not configure a gateway in the VLAN, the software uses the globally configured gateway and gives the gateway a metric value of 1.

To configure a designated management VLAN, enter commands such as the following.

```
Brocade(config)# vlan 10 by port
Brocade(config-vlan-10)# untag ethernet 1/1/1 to 1/1/4
Brocade(config-vlan-10)# management-vlan
Brocade(config-vlan-10)# default-gateway 10.10.10.1 1
Brocade(config-vlan-10)# default-gateway 10.20.20.1 2
```

These commands configure port-based VLAN 10 to consist of ports 1/1/1–1/1/4 and to be the designated management VLAN. The last two commands configure default gateways for the VLAN. Since the 10.10.10.1 gateway has a lower metric, the software uses this gateway. The other gateway remains in the configuration but is not used. You can use the other one by changing the metrics so that the 10.20.20.1 gateway has the lower metric.

Syntax: [**no**] **default-gateway** *ip-addr metric*

The *ip-addr* parameters specify the IP address of the gateway router.

The *metric* parameter specifies the metric (cost) of the gateway. You can specify a value from 1–5. There is no default. The software uses the gateway with the lowest metric.

Device management security

By default, all management access is disabled. Each of the following management access methods must be specifically enabled as required in your installation:

- SSHv2
- SNMP

The commands for granting access to each of these management interfaces is described in the following.

Allowing SSHv2 access to the Brocade device

To allow SSHv2 access to the Brocade device, you must generate a Crypto Key as shown in the following command.

```
Brocade(config)# crypto key generate
```

Syntax: **crypto key** [**generate** | **zeroize**]

The **generate** parameter generates a dsa key pair.

The **zeroize** parameter deletes the currently operative dsa key pair.

In addition, you must use AAA authentication to create a password to allow SSHv2 access. For example the following command configures AAA authentication to use TACACS+ for authentication as the default or local if TACACS+ is not available.

```
Brocade(config)# aaa authentication login default tacacs+ local
```

Allowing SNMP access to the Brocade device

To allow SNMP access to the Brocade device, enter the following command.

```
Brocade(config)# snmp-server
```

Syntax: [no] **snmp-server**

Disabling specific access methods

You can specifically disable the following access methods:

- Telnet access
- SNMP access
- TFTP

NOTE
If you disable Telnet access, you will not be able to access the CLI except through a serial connection to the management module. If you disable SNMP access, you will not be able to use an SNMP-based management applications.

Disabling Telnet access

You can use a Telnet client to access the CLI on the device over the network. If you do not plan to use the CLI over the network and want to disable Telnet access to prevent others from establishing CLI sessions with the device, enter the following command.

```
Brocade(config)# no telnet server
```

To re-enable Telnet operation, enter the following command.

```
Brocade(config)# telnet server
```

Syntax: [no] **telnet server**

Disabling SNMP access

To disable SNMP management of the device.

```
Brocade(config)# no snmp-server
```

To later re-enable SNMP management of the device.

```
Brocade(config)# snmp-server
```

Syntax: **no snmp-server**

Disabling TFTP access

You can globally disable TFTP to block TFTP client access. By default, TFTP client access is enabled.

To disable TFTP client access, enter the following command at the Global CONFIG level of the CLI.

```
Brocade(config)# tftp disable
```

When TFTP is disabled, you are prohibited from using the **copy tftp** command to copy files to the system flash. If you enter this command while TFTP is disabled, the system will reject the command and display an error message.

To re-enable TFTP client access once it is disabled, enter the following command.

```
Brocade(config)# no tftp disable
```

Syntax: [**no**] **tftp disable**

Passwords used to secure access

Passwords can be used to secure the following access methods:

- Telnet access can be secured by setting a Telnet password. Refer to "Setting a Telnet password" on page 13.
- Access to the Privileged EXEC and CONFIG levels of the CLI can be secured by setting passwords for management privilege levels. Refer to "Setting passwords for management privilege levels" on page 14.

This section also provides procedures for enhancing management privilege levels, recovering from a lost password, and disabling password encryption.

NOTE
You also can configure up to 16 user accounts consisting of a user name and password, and assign each user account a management privilege level. Refer to "Local user accounts" on page 17.

Setting a Telnet password

By default, the device does not require a user name or password when you log in to the CLI using Telnet. You can assign a password for Telnet access using one of the following methods.

Set the password "letmein" for Telnet access to the CLI using the following command at the global CONFIG level.

```
Brocade(config)# enable telnet password letmein
```

Syntax: [**no**] **enable telnet password** *string*

Suppressing Telnet connection rejection messages

By default, if a Brocade device denies Telnet management access to the device, the software sends a message to the denied Telnet client. You can optionally suppress the rejection message. When you enable the option, a denied Telnet client does not receive a message from the Brocade device. Instead, the denied client simply does not gain access.

To suppress the connection rejection message, use the following CLI method.

To suppress the connection rejection message sent by the device to a denied Telnet client, enter the following command at the global CONFIG level of the CLI.

```
Brocade(config)# telnet server suppress-reject-message
```

Syntax: [no] **telnet server suppress-reject-message**

Setting passwords for management privilege levels

You can set one password for each of the following management privilege levels:

- **Super User level** – Allows complete read-and-write access to the system. This is generally for system administrators and is the only management privilege level that allows you to configure passwords.
- **Port Configuration level** – Allows read-and-write access for specific ports but not for global (system-wide) parameters.
- **Read Only level** – Allows access to the Privileged EXEC mode and User EXEC mode of the CLI but only with read access.

You can assign a password to each management privilege level. You also can configure up to 16 user accounts consisting of a user name and password, and assign each user account to one of the three privilege levels. Refer to "Local user accounts" on page 17.

NOTE
You must use the CLI to assign a password for management privilege levels.

If you configure user accounts in addition to privilege level passwords, the device will validate a user access attempt using one or both methods (local user account or privilege level password), depending on the order you specify in the authentication-method lists. Refer to "Authentication-method lists" on page 58.

Follow the steps given below to set passwords for management privilege levels.

1. At the opening CLI prompt, enter the following command to change to the Privileged level of the EXEC mode.

```
Brocade> enable
Brocade#
```

2. Access the CONFIG level of the CLI by entering the following command.

```
Brocade# configure terminal
Brocade(config)#
```

3. Enter the following command to set the Super User level password.

```
Brocade(config)# enable super-user-password text
```

NOTE
You must set the Super User level password before you can set other types of passwords. The Super User level password can be an alphanumeric string, but cannot begin with a number.

4. Enter the following commands to set the Port Configuration level and Read Only level passwords.

```
Brocade(config)# enable port-config-password text
Brocade(config)# enable read-only-password text
```

Syntax: **enable super-user-password** *text*

Syntax: **enable port-config-password** *text*

Syntax: **enable read-only-password** *text*

NOTE
If you forget your Super User level password, refer to "Recovering from a lost password" on page 16.

Augmenting management privilege levels

Each management privilege level provides access to specific areas of the CLI by default:

- Super User level provides access to all commands and displays.
- Port Configuration level gives access to:
  - The User EXEC and Privileged EXEC levels
  - The port-specific parts of the CONFIG level
  - All interface configuration levels
- Read Only level gives access to:
  - The User EXEC and Privileged EXEC levels

You can grant additional access to a privilege level on an individual command basis. To grant the additional access, you specify the privilege level you are enhancing, the CLI level that contains the command, and the individual command.

NOTE
This feature applies only to management privilege levels on the CLI.

Enhance the Port Configuration privilege level so users also can enter IP commands at the global CONFIG level.

```
Brocade(config)# privilege configure level 4 ip
```

In this command, **configure** specifies that the enhanced access is for a command at the global CONFIG level of the CLI. The **level 4** parameter indicates that the enhanced access is for management privilege level 4 (Port Configuration). All users with Port Configuration privileges will have the enhanced access. The **ip** parameter indicates that the enhanced access is for the IP commands. Users who log in with valid Port Configuration level user names and passwords can enter commands that begin with "ip" at the global CONFIG level.

Syntax: [**no**] **privilege** *cli-level* **level** *privilege-level command-string*

The *cli-level* parameter specifies the CLI level and can be one of the following values:

- **exec** – EXEC level; for example, Brocade> or Brocade#
- **configure** – CONFIG level; for example, Brocade(config)#
- **interface** – Interface level; for example, Brocade(config-if-6)#
- **loopback-interface** – loopback interface level
- **virtual-interface** – Virtual-interface level; for example, Brocade(config-vif-6)#
- **dot1x** – 802.1X configuration level
- **ipv6-access-list** – IPv6 access list configuration level
- **rip-router** – RIP router level; for example, Brocade(config-rip-router)#
- **ospf-router** – OSPF router level; for example, Brocade(config-ospf-router)#
- **pim-router** – PIM router level; for example, Brocade(config-pim-router)#

- **bgp-router** – BGP4 router level; for example, Brocade(config-bgp-router)#
- **vrrp-router** – VRRP configuration level
- **trunk** – trunk configuration level
- **port-vlan** – Port-based VLAN level; for example, Brocade(config-vlan)#
- **protocol-vlan** – Protocol-based VLAN level

The *privilege-level* indicates the number of the management privilege level you are augmenting. You can specify one of the following:

- **0** – Super User level (full read-write access)
- **4** – Port Configuration level
- **5** – Read Only level

The *command-string* parameter specifies the command you are allowing users with the specified privilege level to enter. To display a list of the commands at a CLI level, enter "?" at that level's command prompt.

Recovering from a lost password

Recovery from a lost password requires direct access to the serial port and a system reset.

NOTE
You can perform this procedure only from the CLI.

Follow the steps given below to recover from a lost password.

1. Start a CLI session over the serial interface to the device.
2. Reboot the device.
3. At the initial boot prompt at system startup, enter **b** to enter the boot monitor mode.
4. Enter **no password** at the prompt. (You cannot abbreviate this command.) This command will cause the device to bypass the system password check.
5. Enter **boot system flash primary** at the prompt.
6. After the console prompt reappears, assign a new password.

Displaying the SNMP community string

If you want to display the SNMP community string, enter the following commands.

```
Brocade(config)# enable password-display
Brocade# show snmp server
```

The **enable password-display** command enables display of the community string, but only in the output of the **show snmp server** command. Display of the string is still encrypted in the startup-config file and running-config. Enter the command at the global CONFIG level of the CLI.

Specifying a minimum password length

By default, the Brocade device imposes no minimum length on the Line (Telnet), Enable, or Local passwords. You can configure the device to require that Line, Enable, and Local passwords be at least a specified length.

For example, to specify that the Line, Enable, and Local passwords be at least 8 characters, enter the following command.

```
Brocade(config)# enable password-min-length 8
```

Syntax: enable password-min-length *number-of-characters*

The *number-of-characters* can be from 1–48.

Local user accounts

You can define up to 16 local user accounts on a Brocade device. User accounts regulate who can access the management functions in the CLI using the following methods:

- Telnet access
- SNMP access

Local user accounts provide greater flexibility for controlling management access to Brocade devices than do management privilege level passwords and SNMP community strings of SNMP versions 1 and 2. You can continue to use the privilege level passwords and the SNMP community strings as additional means of access authentication. Alternatively, you can choose not to use local user accounts and instead continue to use only the privilege level passwords and SNMP community strings. Local user accounts are backward-compatible with configuration files that contain privilege level passwords. Refer to "Setting passwords for management privilege levels" on page 14.

If you configure local user accounts, you also need to configure an authentication-method list for Telnet access and SNMP access. Refer to "Authentication-method lists" on page 58.

For each local user account, you specify a user name. You also can specify the following parameters:

- A password
- A management privilege level, which can be one of the following:
  - **Super User level (default)** – Allows complete read-and-write access to the system. This is generally for system administrators and is the only privilege level that allows you to configure passwords.
  - **Port Configuration level** – Allows read-and-write access for specific ports but not for global parameters.
  - **Read Only level** – Allows access to the Privileged EXEC mode and User EXEC mode with read access only.
- You can set additional username and password rules. Refer to "Enhancements to username and password".

Enhancements to username and password

This section describes the enhancements to the username and password features introduced in earlier releases.

The following rules are enabled by default:

- Users are required to accept the message of the day.

- Users are locked out (disabled) if they fail to login after three attempts. This feature is automatically enabled. Use the **disable-on-login-failure** command to change the number of login attempts (up to 10) before users are locked out.

The following rules are disabled by default:

- Enhanced user password combination requirements
- User password masking
- Quarterly updates of user passwords
- You can configure the system to store up to 15 previously configured passwords for each user.
- You can use the **disable-on-login-failure** command to change the number of login attempts (up to 10) before users are locked out.
- A password can now be set to expire.

Enabling enhanced user password combination requirements

When strict password enforcement is enabled on the Brocade device, you must enter a minimum of eight characters containing the following combinations when you create an enable and a user password:

- At least two upper case characters
- At least two lower case characters
- At least two numeric characters
- At least two special characters

NOTE
Password minimum and combination requirements are strictly enforced.

Use the **enable strict-password-enforcement** command to enable the password security feature.

```
Brocade(config)# enable strict-password-enforcement
```

Syntax: [no] **enable strict-password-enforcement**

This feature is disabled by default.

The following security upgrades apply to the **enable strict-password-enforcement** command:

- Passwords must not share four or more concurrent characters with any other password configured on the router. If the user tries to create a password with four or more concurrent characters, the following error message will be returned.

  ```
  Error - The substring <str> within the password has been used earlier, please
  choose a different password.
  ```

  For example, the previous password was Ma!i4aYa&, the user cannot use any of the following as his or her new password:

  - Ma!imai$D because “Mail” were used consecutively in the previous password
  - &3B9aYa& because “aYa&” were used consecutively in the previous password
  - i4aYEv#8 because “i4aY“were used consecutively in the previous password
- If the user tries to configure a password that was previously used, the Local User Account configuration will not be allowed and the following message will be displayed.

```
This password was used earlier for same or different user, please choose a
different password.
```

Enabling user password masking

By default, when you use the CLI to create a user password, the password displays on the console as you type it. For enhanced security, you can configure the Brocade device to mask the password characters entered at the CLI. When password masking is enabled, the CLI displays asterisks (*) on the console instead of the actual password characters entered.

The following shows the default CLI behavior when configuring a username and password.

```
Brocade(config)# username kelly password summertime
```

The following shows the CLI behavior when configuring a username and password when **password-masking** is enabled.

```
Brocade(config)# username kelly password
Enter Password: ********
```

NOTE
When password masking is enabled, press the [Enter] key before entering the password.

Syntax: username *name* **password** [**Enter**]

For [Enter], press the Enter key. Enter the password when prompted.

If **strict-password-enforcement** is enabled, enter a password which contains the required character combination. Refer to "Enabling enhanced user password combination requirements" on page 18.

To enable password masking, enter the following command.

```
Brocade(config)# enable user password-masking
```

Syntax: [**no**] **enable user password-masking**

Enabling user password aging

For enhanced security, password aging enforces quarterly updates of all user passwords. After 180 days, the CLI will automatically prompt users to change their passwords when they attempt to sign on.

When password aging is enabled, the software records the system time that each user password was configured or last changed. The time displays in the output of the **show running configuration** command, indicated by set-time *time*.

Example

```
Brocade# show run
Current configuration:
....
username waldo password .....
username raveen set-time 2086038248
....
```

The password aging feature uses the SNTP server clock to record the set-time. If the network does not have an SNTP server, then set-time will appear as **set-time 0** in the output of the **show running configuration** command.

A username set-time configuration is removed when:

- The username and password is deleted from the configuration
- The username password expires

When a username set-time configuration is removed, it no longer appears in the **show running configuration** output.

Note that if a username does not have an assigned password, the username will not have a set-time configuration.

Password aging is disabled by default. To enable it, enter the following command at the global CONFIG level of the CLI.

```
Brocade(config)# enable user password-aging
```

Syntax: [**no**] **enable user password-aging**

Configuring password history

By default, the Brocade device stores the last five user passwords for each user. When changing a user password, the user cannot use any of the five previously configured passwords.

For security purposes, you can configure the Brocade device to store up to 15 passwords for each user, so that users do not use the same password multiple times. If a user attempts to use a password that is stored, the system will prompt the user to choose a different password.

To configure enhanced password history, enter a command such as the following at the global CONFIG level of the CLI.

```
Brocade(config)# enable user password-history 15
```

Syntax: [**no**] **enable user password-history** *1 – 15*

Enhanced login lockout

The CLI provides up to three login attempts. If a user fails to login after three attempts, that user is locked out (disabled). If desired, you can increase or decrease the number of login attempts before the user is disabled. To do so, enter a command such as the following at the global CONFIG level of the CLI.

```
Brocade(config)# enable user disable-on-login-failure 7
```

Syntax: enable user disable-on-login-failure *1 – 10*

To re-enable a user that has been locked out, do one of the following:

- Reboot the Brocade device to re-enable all disabled users.
- Enable the user by entering the following command.

  ```
  Brocade(config)# username sandy enable
  ```

Example

```
Brocade(config)# user sandy enable
Brocade# show user
Username  Password                          Encrypt   Priv Status   Expire Time
==========================================================================
==

sandy     $1$Gz...uX/$wQ44fVGtsqbKWkQknzAZ6. enabled   0   enabled  90 days
```

Syntax: **username** *name* **enable**

Setting passwords to expire

You can set a user password to expire. Once a password expires, the administrator must assign a new password to the user. To configure a user password to expire, enter the following.

```
Brocade(config)# username sandy expires 20
```

Syntax: **username** *name* **expires** *days*

Enter 1–365 for number of days. The default is 90 days.

Example

```
Brocade(config)# username sandy expires 20
Brocade# show user
Username     Password                          Encrypt   Priv  Status   Expire
Time
==========================================================================
==
sandy       $1$Gz...uX/$wQ44fVGtsqbKWkQknzAZ6. enabled    0    enabled  20 days
```

Requirement to accept the message of the day

If a message of the day (MOTD) is configured, a user will be required to press the Enter key before he or she can login. MOTD is configured using the **banner motd** command.

There are no new CLI commands for this feature.

NOTE
This requirement is disabled by default, unless configured. Users are not required to press Enter after the MOTD banner is displayed. Refer to *Brocade ICX 6650 Administration Guide*.

Local user account configuration

You can create accounts for local users with or without passwords. Accounts with passwords can have encrypted or unencrypted passwords.

You can assign privilege levels to local user accounts, but on a new device, you must create a local user account that has a Super User privilege before you can create accounts with other privilege levels.

NOTE
You must grant Super User level privilege to at least one account before you add accounts with other privilege levels. You need the Super User account to make further administrative changes.

Local user accounts with no passwords

To create a user account without a password, enter the following command at the global CONFIG level of the CLI.

```
Brocade(config)# username wonka nopassword
```

Syntax: [**no**] **username** *user-string* **privilege** *privilege-level* **nopassword**

Local user accounts with unencrypted passwords

If you want to use unencrypted passwords for local user accounts, enter a command such as the following at the global CONFIG level of the CLI.

```
Brocade(config)# username wonka password willy
```

If password masking is enabled, press the [Enter] key before entering the password.

```
Brocade(config)# username wonka
Enter Password:  willy
```

The above commands add a local user account with the user name “wonka” and the password “willy”. This account has the Super User privilege level; this user has full access to all configuration and display features.

```
Brocade(config)# username waldo privilege 5 password whereis
```

This command adds a user account for user name “waldo”, password “whereis”, with the Read Only privilege level. Waldo can look for information but cannot make configuration changes.

Syntax: [**no**] **username** *user-string* **privilege** *privilege-level* **password** | **nopassword** *password-string*

You can enter up to 48 characters for *user-string*.

The **privilege** *privilege-level* parameter specifies the privilege level for the account. You can specify one of the following:

- **0** – Super User level (full read-write access)
- **4** – Port Configuration level
- **5** – Read Only level

The default privilege level is **0**. If you want to assign Super User level access to the account, you can enter the command without **privilege 0**, as shown in the command example above.

The **password** | **nopassword** parameter indicates whether the user must enter a password. If you specify **password**, enter the string for the user's password. You can enter up to 48 characters for *password-string*. If **strict password enforcement** is enabled on the device, you must enter a minimum of eight characters containing the following combinations:

- At least two upper case characters
- At least two lower case characters
- At least two numeric characters

- At least two special characters

NOTE
You must be logged on with Super User access (privilege level 0) to add user accounts or configure other access parameters.

To display user account information, enter the following command.

```
Brocade# show users
```

Syntax: show users

Local accounts with encrypted passwords

You can create local user accounts with MD5 encrypted passwords using one of the following methods:

- Issuing the **service password-encryption** command after creating the local user account with a **username** *user-string* [**privilege** *privilege-level*] **password 0** command
- Using the **username** *user-string* **create-password** command

  NOTE
  To create an encrypted all-numeric password, use the **username** *user-string* **create-password** command.

If you create a local user account using the commands discussed in "Local user accounts with unencrypted passwords" on page 22, you can issue the **service password-encryption** command to encrypt all passwords that have been previously entered.

Example

```
Brocade(config)# username wonka privilege 5 password willy
Brocade(config)# service password-encryption
```

Creating a password option

As an alternative to the commands above, the **create-password** option allows you to create an encrypted password in one line of command. Also, this new option allows you to create an all-numeric, encrypted password.

You can enter.

```
Brocade(config)# username wonka privilege 5 create-password willy
```

Syntax: [**no**] **username** *user-string* [**privilege** *privilege-level*] **create-password** *password-string*

You can enter up to 48 characters for *user-string*. This string can be alphanumeric or all-numeric.

The **privilege** parameter specifies the privilege level for the account. You can specify one of the following:

- **0** – Super User level (full read-write access)
- **4** – Port Configuration level
- **5** – Read Only level

Enter up to 255 alphanumeric characters for *password-string*.

Changing a local user password

To change a local user password for an existing local user account, enter a command such as the following at the global CONFIG level of the CLI.

NOTE
You must be logged on with Super User access (privilege level 0) to change user passwords.

```
Brocade(config)# username wonka password willy
```

If password masking is enabled, enter the username, press the [Enter] key, then enter the password.

```
Brocade(config)# username wonka password
Enter Password:  willy
```

The above commands change wonka's user name password to “willy”.

Syntax: [**no**] **username** *user-string* **password** *password-string*

Enter up to 48 characters for *user-string*.

The *password-string* parameter is the user password. The password can be up to 48 characters and must differ from the current password and two previously configured passwords.

When a password is changed, a message such as the following is sent to the Syslog.

```
SYSLOG: <14>Jan 1 00:00:00 10.44.9.11 Security: Password has been changed for user
tester from console session.
```

The message includes the name of the user whose password was changed and during which session type, such as Console, Telnet, SSH, SNMP, or others, the password was changed.

TACACS and TACACS+ security

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

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

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

How TACACS+ differs from TACACS

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

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

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

TACACS/TACACS+ authentication, authorization, and accounting

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

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

NOTE
By default, a user logging into the device from Telnet or SSH would first enter the User EXEC level. The user can enter the **enable** command to get to the Privileged EXEC level.

A user that is successfully authenticated can be automatically placed at the Privileged EXEC level after login. Refer to "Entering privileged EXEC mode after a Telnet or SSH login" on page 35.

Configuring TACACS/TACACS+ for devices in a Brocade IronStack

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

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

```
aaa authentication login default tacacs+ enable
aaa authentication login privilege-mode
aaa authorization commands 0 default  tacacs+
aaa authorization exec default  tacacs+
aaa accounting commands 0 default start-stop  tacacs+
aaa accounting exec default start-stop  tacacs+
aaa accounting system default start-stop  tacacs+
enable aaa console
hostname Fred
ip address 10.10.6.56/255
tacacs-server host 255.253.255
tacacs-server key 1 $Gsig@U\
```

kill console

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

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

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

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

In case a user forgets to log out or a console is left unattended, you can also configure the console timeout (in minutes) on all stack units (including the Active Controller).

```
Brocade(config)# stack unit 3
Brocade(config-unit-3)# console timeout 5
Brocade(config-unit-3)# exit
Brocade(config)# stack unit 4
Brocade(config-unit-4)# console timeout 5
```

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

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

stack9# show telnet
Console connections (by unit number):
 1      established
        you are connecting to this session
        1 minutes 5 seconds in idle
 2      established
        1 hours 4 minutes 18 seconds in idle
 3      established
        1 hours 4 minutes 15 seconds in idle
 4      established
        1 hours 4 minutes 9 seconds in idle
```

```
Telnet connections (inbound):
 1      closed
 2      closed
 3      closed
 4      closed
 5      closed
Telnet connection (outbound):
 6      closed
SSH connections:
 1      closed
 2      closed
 3      closed
 4      closed
 5      closed
stack9#
```

TACACS authentication

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

When TACACS authentication takes place, the following events occur.

1. A user attempts to gain access to the Brocade device by doing one of the following:
   - Logging into the device using Telnet or SSH
   - Entering the Privileged EXEC level or CONFIG level of the CLI
2. The user is prompted for a username and password.
3. The user enters a username and password.
4. The Brocade device sends a request containing the username and password to the TACACS server.
5. The username and password are validated in the TACACS server database.
6. If the password is valid, the user is authenticated.

TACACS+ authentication

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

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

8. The password is validated in the TACACS+ server database.
9. If the password is valid, the user is authenticated.

TACACS+ authorization

Brocade devices support two kinds of TACACS+ authorization:

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

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

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

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

1. A Telnet or SSH user previously authenticated by a TACACS+ server enters a command on the Brocade device.
2. The Brocade device looks at its configuration to see if the command is at a privilege level that requires TACACS+ command authorization.
3. If the command belongs to a privilege level that requires authorization, the Brocade device consults the TACACS+ server to see if the user is authorized to use the command.
4. If the user is authorized to use the command, the command is executed.

TACACS+ accounting

TACACS+ accounting works as follows.

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

AAA operations for TACACS/TACACS+

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

TABLE 3 AAA operations

User action	Applicable AAA operations
User attempts to gain access to the Privileged EXEC and CONFIG levels of the CLI	Enable authentication: aaa authentication enable default *method-list*
	Exec authorization (TACACS+): aaa authorization exec default tacacs+
	System accounting start (TACACS+): aaa accounting system default start-stop *method-list*
User logs in using Telnet/SSH	Login authentication: aaa authentication login default *method-list*
	Exec authorization (TACACS+): aaa authorization exec default tacacs+
	Exec accounting start (TACACS+): aaa accounting exec default *method-list* System accounting start (TACACS+): aaa accounting system default start-stop *method-list*
User logs out of Telnet/SSH session	Command accounting (TACACS+): aaa accounting commands *privilege-level* default start-stop *method-list* EXEC accounting stop (TACACS+): aaa accounting exec default start-stop *method-list*
User enters system commands (for example, **reload**, **boot system**)	Command authorization (TACACS+): aaa authorization commands *privilege-level* default *method-list*
	Command accounting (TACACS+): aaa accounting commands *privilege-level* default start-stop *method-list* System accounting stop (TACACS+): aaa accounting system default start-stop *method-list*
User enters the command: [no] aaa accounting system default start-stop *method-list*	Command authorization (TACACS+): aaa authorization commands *privilege-level* default *method-list*
	Command accounting (TACACS+): aaa accounting commands *privilege-level* default start-stop *method-list* System accounting start (TACACS+): aaa accounting system default start-stop *method-list*
User enters other commands	Command authorization (TACACS+): aaa authorization commands *privilege-level* default *method-list*
	Command accounting (TACACS+): aaa accounting commands *privilege-level* default start-stop *method-list*

AAA security for commands pasted into the running-config

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

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

TACACS/TACACS+ configuration considerations

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

Configuring TACACS

Follow the procedure given below for TACACS configurations.

1. Identify TACACS servers. Refer to "Identifying the TACACS/TACACS+ servers" on page 31.
2. Set optional parameters. Refer to "Setting optional TACACS and TACACS+ parameters" on page 32.
3. Configure authentication-method lists. Refer to "Configuring authentication-method lists for TACACS and TACACS+" on page 34.

Configuring TACACS+

Follow the procedure given below for TACACS+ configurations.

1. Identify TACACS+ servers. Refer to "Identifying the TACACS/TACACS+ servers" on page 31.
2. Set optional parameters. Refer to "Setting optional TACACS and TACACS+ parameters" on page 32.
3. Configure authentication-method lists. Refer to "Configuring authentication-method lists for TACACS and TACACS+" on page 34.
4. Optionally configure TACACS+ authorization. Refer to "Configuring TACACS+ authorization" on page 36.
5. Optionally configure TACACS+ accounting. Refer to "TACACS+ accounting configuration" on page 39.

Enabling TACACS

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

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

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

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

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

Identifying the TACACS/TACACS+ servers

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

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

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

Syntax: **tacacs-server host** *ip-addr* | *ipv6-addr* | *hostname* [**auth-port** *umber*]

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

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

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

1. 10.94.6.161
2. 10.94.6.191
3. 10.94.6.122

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

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

NOTE
If you erase a **tacacs-server** command (by entering "no" followed by the command), make sure you also erase the **aaa** commands that specify TACACS/TACACS+ as an authentication method. (Refer to "Configuring authentication-method lists for TACACS and TACACS+" on page 34.) Otherwise, when you exit from the CONFIG mode or from a Telnet session, the system continues to believe it is TACACS/TACACS+ enabled and you will not be able to access the system.

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

Specifying different servers for individual AAA functions

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

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

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

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

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

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

Setting optional TACACS and TACACS+ parameters

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

- **TACACS+ key** – This parameter specifies the value that the Brocade device sends to the TACACS+ server when trying to authenticate user access.
- **Retransmit interval** – This parameter specifies how many times the Brocade device will resend an authentication request when the TACACS/TACACS+ server does not respond. The retransmit value can be from 1 – 5 times. The default is 3 times.
- **Dead time** – This parameter specifies how long the Brocade device waits for the primary authentication server to reply before deciding the server is dead and trying to authenticate using the next server. The dead-time value can be from 1 – 5 seconds. The default is 3 seconds.
- **Timeout** – This parameter specifies how many seconds the Brocade device waits for a response from a TACACS/TACACS+ server before either retrying the authentication request, or determining that the TACACS/TACACS+ servers are unavailable and moving on to the next authentication method in the authentication-method list. The timeout can be from 1 – 15 seconds. The default is 3 seconds.

Setting the TACACS+ key

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

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

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

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

Syntax: tacacs-server key [**0** | **1**] *string*

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

```
Brocade(config)# tacacs-server key 1 abc
Brocade(config)# write terminal
...
tacacs-server host 10.2.3.5 auth-port 49
tacacs key 1 $!2d
```

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

Setting the retransmission limit

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

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

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

Syntax: tacacs-server retransmit *number*

Setting the timeout parameter

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

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

Syntax: tacacs-server timeout *number*

Configuring authentication-method lists for TACACS and TACACS+

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

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

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

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

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

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

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

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

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

Syntax: [**no**] **aaa authentication enable** | **login default** *method1* [*method2*] [*method3*] [*method4*] [*method5*] [*method6*] [*method7*]

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

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

TABLE 4 Authentication method values

Method parameter	Description
line	Authenticate using the password you configured for Telnet access. The Telnet password is configured using the **enable telnet password...** command. Refer to "Setting a Telnet password" on page 13.
enable	Authenticate using the password you configured for the Super User privilege level. This password is configured using the **enable super-user-password...** command. Refer to "Setting passwords for management privilege levels" on page 14.

TABLE 4 Authentication method values (Continued)

Method parameter	Description
local	Authenticate using a local user name and password you configured on the device. Local user names and passwords are configured using the **username...** command. Refer to "Local user account configuration" on page 21.
tacacs	Authenticate using the database on a TACACS server. You also must identify the server to the device using the **tacacs-server** command.
tacacs+	Authenticate using the database on a TACACS+ server. You also must identify the server to the device using the **tacacs-server** command.
radius	Authenticate using the database on a RADIUS server. You also must identify the server to the device using the **radius-server** command.
none	Do not use any authentication method. The device automatically permits access.

NOTE
For examples of how to define authentication-method lists for types of authentication other than TACACS/TACACS+, refer to "Authentication-method lists" on page 58.

Entering privileged EXEC mode after a Telnet or SSH login

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

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

Syntax: aaa authentication login privilege-mode

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

Configuring enable authentication to prompt for password only

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

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

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

Syntax: [no] aaa authentication enable implicit-user

Telnet and SSH prompts when the TACACS+ server is unavailable

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

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

Configuring TACACS+ authorization

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

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

Configuring EXEC authorization

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

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

Syntax: aaa authorization exec default tacacs+ | none

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

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

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

Also note that in order for the **aaa authorization exec default tacacs+** command to work, either the **aaa authentication enable default tacacs+** command, or the **aaa authentication login privilege-mode** command must also exist in the configuration.

Configuring an Attribute-Value pair on the TACACS+ server

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

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

Example

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

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

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

Example

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

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

TABLE 5 Brocade equivalents for non-“foundry-privlvl” A-V pair values

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

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

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

Example

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

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

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

Configuring command authorization

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

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

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

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

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

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

NOTE
TACACS+ command authorization can be performed only for commands entered from Telnet or SSH sessions, or from the console.

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

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

If configured, command accounting is performed for these commands.

AAA support for console commands

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

- Login prompt that uses AAA authentication, using authentication-method lists

- Exec Authorization
- Exec Accounting
- Command authorization
- Command accounting
- System accounting

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

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

Syntax: [no] **enable aaa console**

TACACS+ accounting configuration

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

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

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

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

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

Configuring TACACS+ accounting for CLI commands

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

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

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

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

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

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

- **0** – Records commands available at the Super User level (all commands)

- 4 – Records commands available at the Port Configuration level (port-config and read-only commands)
- 5 – Records commands available at the Read Only level (read-only commands)

Configuring TACACS+ accounting for system events

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

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

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

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

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

You can designate the lowest-numbered IP address configured an Ethernet port, loopback interface, or virtual interface as the source IP address for all TACACS/TACACS+ packets from the Layer 3 switch. For configuration details, refer to Brocade ICX 6650 *Layer 3 Routing Configuration Guide*.

Displaying TACACS/TACACS+ statistics and configuration information

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

```
Brocade# show aaa
Tacacs+ key: foundry
Tacacs+ retries: 1
Tacacs+ timeout: 15 seconds
Tacacs+ dead-time: 3 minutes
Tacacs+ Server: 10.95.6.90 Port:49:
                opens=6 closes=3 timeouts=3 errors=0
                packets in=4 packets out=4
no connection
Radius key: networks
Radius retries: 3
Radius timeout: 3 seconds
Radius dead-time: 3 minutes
Radius Server: 10.95.6.90 Auth Port=1645 Acct Port=1646:
                opens=2 closes=1 timeouts=1 errors=0
                packets in=1 packets out=4
no connection
```

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

TABLE 6 Output of the **show aaa** command for TACACS/TACACS+

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

RADIUS security

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

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

RADIUS authentication, authorization, and accounting

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

RADIUS authentication

When RADIUS authentication takes place, the following events occur.

1. A user attempts to gain access to the Brocade device by doing one of the following:
   - Logging into the device using Telnet or SSH
   - Entering the Privileged EXEC level or CONFIG level of the CLI
2. The user is prompted for a username and password.
3. The user enters a username and password.

4. The Brocade device sends a RADIUS Access-Request packet containing the username and password to the RADIUS server.
5. The RADIUS server validates the Brocade device using a shared secret (the RADIUS key).
6. The RADIUS server looks up the username in its database.
7. If the username is found in the database, the RADIUS server validates the password.
8. If the password is valid, the RADIUS server sends an Access-Accept packet to the Brocade device, authenticating the user. Within the Access-Accept packet are three Brocade vendor-specific attributes that indicate:
   - The privilege level of the user
   - A list of commands
   - Whether the user is allowed or denied usage of the commands in the list

   The last two attributes are used with RADIUS authorization, if configured.
9. The user is authenticated, and the information supplied in the Access-Accept packet for the user is stored on the Brocade device. The user is granted the specified privilege level. If you configure RADIUS authorization, the user is allowed or denied usage of the commands in the list.

RADIUS authorization

When RADIUS authorization takes place, the following events occur.

1. A user previously authenticated by a RADIUS server enters a command on the Brocade device.
2. The Brocade device looks at its configuration to see if the command is at a privilege level that requires RADIUS command authorization.
3. If the command belongs to a privilege level that requires authorization, the Brocade device looks at the list of commands delivered to it in the RADIUS Access-Accept packet when the user was authenticated. (Along with the command list, an attribute was sent that specifies whether the user is permitted or denied usage of the commands in the list.)

   NOTE
   After RADIUS authentication takes place, the command list resides on the Brocade device. The RADIUS server is not consulted again once the user has been authenticated. This means that any changes made to the user command list on the RADIUS server are not reflected until the next time the user is authenticated by the RADIUS server, and the new command list is sent to the Brocade device.
4. If the command list indicates that the user is authorized to use the command, the command is executed.

RADIUS accounting

RADIUS accounting works as follows.

1. One of the following events occur on the Brocade device:
   - A user logs into the management interface using Telnet or SSH
   - A user enters a command for which accounting has been configured
   - A system event occurs, such as a reboot or reloading of the configuration file

2. The Brocade device checks its configuration to see if the event is one for which RADIUS accounting is required.
3. If the event requires RADIUS accounting, the Brocade device sends a RADIUS Accounting Start packet to the RADIUS accounting server, containing information about the event.
4. The RADIUS accounting server acknowledges the Accounting Start packet.
5. The RADIUS accounting server records information about the event.
6. When the event is concluded, the Brocade device sends an Accounting Stop packet to the RADIUS accounting server.
7. The RADIUS accounting server acknowledges the Accounting Stop packet.

AAA operations for RADIUS

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

TABLE 7 AAA operations for RADIUS

User action	Applicable AAA operations
User attempts to gain access to the Privileged EXEC and CONFIG levels of the CLI	Enable authentication: aaa authentication enable default *method-list*
	System accounting start: aaa accounting system default start-stop *method-list*
User logs in using Telnet/SSH	Login authentication: aaa authentication login default *method-list*
	EXEC accounting Start: aaa accounting exec default start-stop *method-list* System accounting Start: aaa accounting system default start-stop *method-list*
User logs out of Telnet/SSH session	Command authorization for **logout** command: aaa authorization commands *privilege-level* default *method-list*
	Command accounting: aaa accounting commands *privilege-level* default start-stop *method-list* EXEC accounting stop: aaa accounting exec default start-stop *method-list*
User enters system commands (for example, **reload**, **boot system**)	Command authorization: aaa authorization commands *privilege-level* default *method-list*
	Command accounting: aaa accounting commands *privilege-level* default start-stop *method-list* System accounting stop: aaa accounting system default start-stop *method-list*
User enters the command: [no] aaa accounting system default start-stop *method-list*	Command authorization: aaa authorization commands *privilege-level* default *method-list*
	Command accounting: aaa accounting commands *privilege-level* default start-stop *method-list* System accounting start: aaa accounting system default start-stop *method-list*

TABLE 7 AAA operations for RADIUS

User action	Applicable AAA operations
User enters other commands	Command authorization: aaa authorization commands *privilege-level* default *method-list*
	Command accounting: aaa accounting commands *privilege-level* default start-stop *method-list*

AAA security for commands pasted Into the running-config

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

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

NOTE
Since RADIUS command authorization relies on a list of commands received from the RADIUS server when authentication is performed, it is important that you use RADIUS authentication when you also use RADIUS command authorization.

RADIUS configuration considerations

- You must deploy at least one RADIUS server in your network.
- Brocade devices support authentication using up to eight RADIUS servers, including those used for 802.1X authentication and for management. The device tries to use the servers in the order you add them to the device configuration. If one RADIUS server times out (does not respond), the Brocade device tries the next one in the list. Servers are tried in the same sequence each time there is a request.
- You can optionally configure a RADIUS server as a ***port server***, indicating that the server will be used only to authenticate users on ports to which it is mapped, as opposed to globally authenticating users on all ports of the device. In earlier releases, all configured RADIUS servers are “global” servers and apply to users on all ports of the device. Refer to “RADIUS server per port” on page 48.
- You can map up to eight RADIUS servers to each port on the Brocade device. The port will authenticate users using only the RADIUS servers to which it is mapped. If there are no RADIUS servers mapped to a port, it will use the “global” servers for authentication. In earlier releases, all RADIUS servers are “global” servers and cannot be bound to individual ports. Refer to “RADIUS server to individual ports mapping” on page 49.
- You can select only one primary authentication method for each type of access to a device (CLI through Telnet, CLI Privileged EXEC and CONFIG levels). For example, you can select RADIUS as the primary authentication method for Telnet CLI access, but you cannot also select TACACS+ authentication as the primary method for the same type of access. However, you can configure backup authentication methods for each access type.

Configuring RADIUS

Follow the procedure given below to configure a Brocade device for RADIUS.

1. Configure Brocade vendor-specific attributes on the RADIUS server. Refer to "Brocade-specific attributes on the RADIUS server" on page 45.
2. Identify the RADIUS server to the Brocade device. Refer to "Identifying the RADIUS server to the Brocade device" on page 47.
3. Optionally specify different servers for individual AAA functions. Refer to "Specifying different servers for individual AAA functions" on page 48.
4. Optionally configure the RADIUS server as a "port only" server. Refer to "RADIUS server per port" on page 48.
5. Optionally bind the RADIUS servers to ports on the Brocade device. Refer to "RADIUS server to individual ports mapping" on page 49.
6. Set RADIUS parameters. Refer to "RADIUS parameters" on page 50.
7. Configure authentication-method lists. Refer to "Setting authentication-method lists for RADIUS" on page 51.
8. Optionally configure RADIUS authorization. Refer to "RADIUS authorization" on page 53.
9. Optionally configure RADIUS accounting. "RADIUS accounting" on page 55.

Brocade-specific attributes on the RADIUS server

NOTE
For all Brocade devices, RADIUS Challenge is supported for 802.1x authentication but not for login authentication.

During the RADIUS authentication process, if a user supplies a valid username and password, the RADIUS server sends an Access-Accept packet to the Brocade device, authenticating the user. Within the Access-Accept packet are three Brocade vendor-specific attributes that indicate:

- The privilege level of the user
- A list of commands
- Whether the user is allowed or denied usage of the commands in the list

You must add these three Brocade vendor-specific attributes to your RADIUS server configuration, and configure the attributes in the individual or group profiles of the users that will access the Brocade device.

Brocade Vendor-ID is 1991, with Vendor-Type 1. The following table describes the Brocade vendor-specific attributes.

TABLE 8 Brocade vendor-specific attributes for RADIUS

Attribute name	Attribute ID	Data type	Description
foundry-privilege-level	1	integer	Specifies the privilege level for the user. This attribute can be set to one of the following: • **0** - S**uper User level** – Allows complete read-and-write access to the system. This is generally for system administrators and is the only management privilege level that allows you to configure passwords. • **4** - **Port Configuration level** – Allows read-and-write access for specific ports but not for global (system-wide) parameters. • **5** - **Read Only level** – Allows access to the Privileged EXEC mode and User EXEC mode of the CLI but only with read access.
foundry-command-string	2	string	Specifies a list of CLI commands that are permitted or denied to the user when RADIUS authorization is configured. The commands are delimited by semi-colons (;). You can specify an asterisk (*) as a wildcard at the end of a command string. For example, the following command list specifies all **show** and **debug ip** commands, as well as the **write terminal** command: show *; debug ip *; write term*
foundry-command-exception-flag	3	integer	Specifies whether the commands indicated by the foundry-command-string attribute are permitted or denied to the user. This attribute can be set to one of the following: • **0** - Permit execution of the commands indicated by foundry-command-string, deny all other commands. • **1** - Deny execution of the commands indicated by foundry-command-string, permit all other commands.
foundry-access-list	5	string	Specifies the access control list to be used for RADIUS authorization. Enter the access control list in the following format. type=string, value="ipacl.[e|s].[in|out] = [<acl-name>|<acl-number>] <separator> macfilter.in = [<acl-name>|<acl-number>] Where: • separator can be a space, newline, semicolon, comma, or null character • ipacl.e is an extended ACL; ipacl.s is a standard ACL.
foundry-MAC-authent-needs-802x	6	integer	Specifies whether or not 802.1x authentication is required and enabled. **0** - Disabled **1** - Enabled

TABLE 8 Brocade vendor-specific attributes for RADIUS (Continued)

Attribute name	Attribute ID	Data type	Description
foundry-802.1x-valid-lookup	7	integer	Specifies if 802.1x lookup is enabled: **0** - Disabled **1** - Enabled
foundry-MAC-based-VLAN-QOS	8	integer	Specifies the priority for MAC-based VLAN QOS: **0** - qos_priority_0 **1** - qos_priority_1 **2** - qos_priority_2 **3** - qos_priority_3 **4** - qos_priority_4 **5** - qos_priority_5 **6** - qos_priority_6 **7** - qos_priority_7

Enabling SNMP to configure RADIUS

To enable SNMP access to RADIUS MIB objects on the device, enter a command such as the following.

```
Brocade(config)# enable snmp config-radius
```

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

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

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

Identifying the RADIUS server to the Brocade device

To use a RADIUS server to authenticate access to a Brocade device, you must identify the server to the Brocade device.

Example

```
Brocade(config)# radius-server host 10.157.22.99
```

Syntax: **radius-server host** *ip-addr* | *iipv6-addr* | *server-name* [**auth-port** *number*] [**acct-port** *number*]

The **host** *ip-addr* | *ipv6-addr* | *server-name* parameter is either an IP address or an ASCII text string.

The *auth-port* parameter is the Authentication port number. The default is 1645.

The *acct-port* parameter is the Accounting port number. The default is 1646.

Specifying different servers for individual AAA functions

In a RADIUS configuration, you can designate a server to handle a specific AAA task. For example, you can designate one RADIUS server to handle authorization and another RADIUS server to handle accounting. You can specify individual servers for authentication and accounting, but not for authorization. You can set the RADIUS key for each server.

To specify different RADIUS servers for authentication, authorization, and accounting, enter commands such as the following.

```
Brocade(config)# radius-server host 10.2.3.4 authentication-only key abc
Brocade(config)# radius-server host 10.2.3.5 authorization-only key def
Brocade(config)# radius-server host 10.2.3.6 accounting-only key ghi
```

Syntax: **radius-server host** *ip-addr* | *ipv6-addr* | *server-name* [**auth-port** *number*] [**acct-port** *number*] [**authentication-only** | **accounting-only** | **default**] [**key 0** | **1** *string*]

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

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

RADIUS server per port

You can optionally configure a RADIUS server per port, indicating that it will be used only to authenticate users on ports to which it is mapped. A RADIUS server that is not explicitly configured as a RADIUS server per port is a ***global server***, and can be used to authenticate users on ports to which no RADIUS servers are mapped.

RADIUS server per port configuration notes

- This feature works with 802.1X and multi-device port authentication only.
- You can define up to eight RADIUS servers per Brocade device.

RADIUS configuration example and command syntax

The following shows an example configuration.

```
Brocade(config)# radius-server host 10.10.10.103 auth-port 1812 acct-port 1813
default key mykeyword dot1x port-only
Brocade(config)# radius-server host 10.10.10.104 auth-port 1812 acct-port 1813
default key mykeyword dot1x port-only
Brocade(config)# radius-server host 10.10.10.105 auth-port 1812 acct-port 1813
default key mykeyword dot1x
Brocade(config)# radius-server host 10.10.10.106 auth-port 1812 acct-port 1813
default key mykeyword dot1x
```

The above configuration has the following affect:

- RADIUS servers 10.10.10.103 and 10.10.10.104 will be used only to authenticate users on ports to which the servers are mapped. To map a RADIUS server to a port, refer to "RADIUS server to individual ports mapping" on page 49.

- RADIUS servers 10.10.10.105 and 10.10.10.106 will be used to authenticate users on ports to which no RADIUS servers are mapped. For example, port e 9, to which no RADIUS servers are mapped, will send a RADIUS request to the first configured RADIUS server, 10.10.10.105. If the request fails, it will go to the second configured RADIUS server, 10.10.10.106. It will not send requests to 10.10.10.103 or 10.10.10.104, since these servers are configured as port servers.

Syntax: **radius-server host** *ip-addr* | *server-name* [**auth-port** *number*] [**acct-port** *number*] [**default key** *string* **dot1x**] [**port-only**]

The **host** *ip-addr* is the IPv4 address.

The **auth-port** *number* parameter is the Authentication port number; it is an optional parameter. The default is 1645.

The **acct-port** *number* parameter is the Accounting port number; it is an optional parameter. The default is 1646.

The **default key** *string* dot1x parameter indicates that this RADIUS server supports the 802.1X standard. A RADIUS server that supports the 802.1X standard can also be used to authenticate non-802.1X authentication requests.

The **port-only** parameter is optional and specifies that the server will be used only to authenticate users on ports to which it is mapped.

RADIUS server to individual ports mapping

You can map up to eight RADIUS servers to each port on the Brocade device. The port will authenticate users using only the RADIUS servers to which the port is mapped. If there are no RADIUS servers mapped to a port, it will use the “global” servers for authentication.

As in previous releases, a port goes through the list of servers in the order in which it was mapped or configured, until a server that can perform the requested function is found, or until every server in the list has been tried.

RADIUS server-to-ports configuration notes

- This feature works with 802.1X and multi-device port authentication only.
- You can map a RADIUS server to a physical port only. You cannot map a RADIUS server to a VE.

RADIUS server-to-ports configuration example and command syntax

To map a RADIUS server to a port, enter commands such as the following.

```
Brocade(config)# int e 3
Brocade(config-if-e1000-3)# dot1x port-control auto
Brocade(config-if-e1000-3)# use-radius-server 10.10.10.103
Brocade(config-if-e1000-3)# use-radius-server 10.10.10.110
```

With the above configuration, port e 3 would send a RADIUS request to 10.10.10.103 first, since it is the first server mapped to the port. If it fails, it will go to 10.10.10.110.

Syntax: **use-radius-server** *ip-addr*

The **host** *ip-addr* is an IPv4 address.

RADIUS parameters

You can set the following parameters in a RADIUS configuration:

- **RADIUS key** – This parameter specifies the value that the Brocade device sends to the RADIUS server when trying to authenticate user access.
- **Retransmit interval** – This parameter specifies how many times the Brocade device will resend an authentication request when the RADIUS server does not respond. The retransmit value can be from 1 – 5 times. The default is 3 times.
- **Timeout** – This parameter specifies how many seconds the Brocade device waits for a response from a RADIUS server before either retrying the authentication request, or determining that the RADIUS servers are unavailable and moving on to the next authentication method in the authentication-method list. The timeout can be from 1 – 15 seconds. The default is 3 seconds.

Setting the RADIUS key

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

To specify a RADIUS server key, enter a command such as the following.

```
Brocade(config)# radius-server key mirabeau
```

Syntax: radius-server key [**0** | **1**] *string*

When you display the configuration of the Brocade device, the RADIUS key is encrypted.

Example

```
Brocade(config)# radius-server key 1 abc
Brocade(config)# write terminal
...
radius-server host 10.2.3.5
radius key 1 $!2d
```

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

Setting the retransmission limit

The **retransmit** parameter specifies the maximum number of retransmission attempts. When an authentication request times out, the Brocade software will retransmit the request up to the maximum number of retransmissions configured. The default retransmit value is 3 retries. The range of retransmit values is from 1 – 5.

To set the RADIUS retransmit limit, enter a command such as the following.

```
Brocade(config)# radius-server retransmit 5
```

Syntax: radius-server retransmit *number*

Setting the timeout parameter

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

```
Brocade(config)# radius-server timeout 5
```

Syntax: radius-server timeout *number*

Setting RADIUS over IPv6

Brocade devices support the ability to send RADIUS packets over an IPv6 network.

To enable the Brocade device to send RADIUS packets over IPv6, enter a command such as the following at the Global CONFIG level of the CLI.

```
Brocade(config)# radius-server host ipv6 3000::300
```

Syntax: radius-server host ipv6 *ipv6-host address*

The *ipv6-host address* is the IPv6 address of the RADIUS server. When you enter the IPv6 host address, you do not need to specify the prefix length. A prefix length of 128 is implied.

Setting authentication-method lists for RADIUS

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

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

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

To create an authentication-method list that specifies RADIUS as the primary authentication method for securing Telnet access to the CLI.

```
Brocade(config)# enable telnet authentication
Brocade(config)# aaa authentication login default radius local
```

The commands above cause RADIUS to be the primary authentication method for securing Telnet access to the CLI. If RADIUS authentication fails due to an error with the server, local authentication is used instead.

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

```
Brocade(config)# aaa authentication enable default radius local none
```

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

Syntax: [**no**] **aaa authentication enable** | **login default** *method1* [*method2*] [*method3*] [*method4*] [*method5*] [*method6*] [*method7*]

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

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

TABLE 9 Authentication method values

Method parameter	Description
line	Authenticate using the password you configured for Telnet access. The Telnet password is configured using the **enable telnet password...** command. Refer to "Setting a Telnet password" on page 13.
enable	Authenticate using the password you configured for the Super User privilege level. This password is configured using the **enable super-user-password...** command. Refer to "Setting passwords for management privilege levels" on page 14.
local	Authenticate using a local user name and password you configured on the device. Local user names and passwords are configured using the **username...** command. Refer to "Local user account configuration" on page 21.
tacacs	Authenticate using the database on a TACACS server. You also must identify the server to the device using the **tacacs-server** command.
tacacs+	Authenticate using the database on a TACACS+ server. You also must identify the server to the device using the **tacacs-server** command.
radius	Authenticate using the database on a RADIUS server. You also must identify the server to the device using the **radius-server** command.
none	Do not use any authentication method. The device automatically permits access.

NOTE
For examples of how to define authentication-method lists for types of authentication other than RADIUS, refer to "Authentication-method lists" on page 58.

Entering privileged EXEC mode after a Telnet or SSH login

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

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

Syntax: aaa authentication login privilege-mode

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

Configuring enable authentication to prompt for password only

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

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

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

Syntax: [no] aaa authentication enable implicit-user

RADIUS authorization

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

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

Configuring EXEC authorization

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

```
Brocade(config)# aaa authorization exec default radius
```

Syntax: aaa authorization exec default radius | none

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

NOTE
If the **aaa authorization exec default radius** command exists in the configuration, following successful authentication the device assigns the user the privilege level specified by the foundry-privilege-level attribute received from the RADIUS server. If the **aaa authorization exec default radius** command does not exist in the configuration, then the value in the foundry-privilege-level attribute is ignored, and the user is granted Super User access.

Also note that in order for the **aaa authorization exec default radius** command to work, either the **aaa authentication enable default radius** command, or the **aaa authentication login privilege-mode** command must also exist in the configuration.

Configuring command authorization

When RADIUS command authorization is enabled, the Brocade device consults the list of commands supplied by the RADIUS server during authentication to determine whether a user can execute a command he or she has entered.

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

```
Brocade(config)# aaa authorization commands 0 default radius
```

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

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

- **0** – Authorization is performed (that is, the Brocade device looks at the command list) for commands available at the Super User level (all commands)
- **4** – Authorization is performed for commands available at the Port Configuration level (port-config and read-only commands)
- **5** – Authorization is performed for commands available at the Read Only level (read-only commands)

NOTE
RADIUS command authorization can be performed only for commands entered from Telnet or SSH sessions, or from the console.

NOTE
Since RADIUS command authorization relies on the command list supplied by the RADIUS server during authentication, you cannot perform RADIUS authorization without RADIUS authentication.

Command authorization and accounting for console commands

The Brocade device supports command authorization and command accounting for CLI commands entered at the console. To configure the device to perform command authorization and command accounting for console commands, enter the following.

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

Syntax: enable aaa console

CAUTION

If you have previously configured the device to perform command authorization using a RADIUS server, entering the enable aaa console command may prevent the execution of any subsequent commands entered on the console.

This happens because RADIUS command authorization requires a list of allowable commands from the RADIUS server. This list is obtained during RADIUS authentication. For console sessions, RADIUS authentication is performed only if you have configured Enable authentication and specified RADIUS as the authentication method (for example, with the aaa authentication enable default radius command). If RADIUS authentication is never performed, the list of allowable commands is never obtained from the RADIUS server. Consequently, there would be no allowable commands on the console.

RADIUS accounting

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

Configuring RADIUS accounting for Telnet/SSH (Shell) access

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

```
Brocade(config)# aaa accounting exec default start-stop radius
```

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

Configuring RADIUS accounting for CLI commands

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

```
Brocade(config)# aaa accounting commands 0 default start-stop radius
```

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

NOTE

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

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

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

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

Configuring RADIUS accounting for system events

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

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

```
Brocade(config)# aaa accounting system default start-stop radius
```

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

Configuring an interface as the source for all RADIUS packets

You can designate the lowest-numbered IP address configured an Ethernet port, loopback interface, or virtual interface as the source IP address for all RADIUS packets from the Layer 3 switch. For configuration details, refer to *Brocade ICX 6650 Layer 3 Routing Configuration Guide*.

Displaying RADIUS configuration information

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

Example

```
Brocade# show aaa
Tacacs+ key: foundry
Tacacs+ retries: 1
Tacacs+ timeout: 15 seconds
Tacacs+ dead-time: 3 minutes
Tacacs+ Server: 10.95.6.90 Port:49:
                opens=6 closes=3 timeouts=3 errors=0
                packets in=4 packets out=4
no connection
Radius key: networks
Radius retries: 3
Radius timeout: 3 seconds
Radius dead-time: 3 minutes
Radius Server: 10.95.6.90 Auth Port=1645 Acct Port=1646:
                opens=2 closes=1 timeouts=1 errors=0
                packets in=1 packets out=4
no connection
```

The following table describes the RADIUS information displayed by the **show aaa** command.

TABLE 10 Output of the **show aaa** command for RADIUS

Field	Description
Radius key	The setting configured with the **radius-server key** command. At the Super User privilege level, the actual text of the key is displayed. At the other privilege levels, a string of periods (....) is displayed instead of the text.
Radius retries	The setting configured with the **radius-server retransmit** command.
Radius timeout	The setting configured with the **radius-server timeout** command.
Radius dead-time	The setting configured with the **radius-server dead-time** command.
Radius Server	For each RADIUS server, the IP address, and the following statistics are displayed: Auth PortRADIUS authentication port number (default 1645) Acct PortRADIUS accounting port number (default 1646) • **opens** - Number of times the port was opened for communication with the server • **closes** - Number of times the port was closed normally • **timeouts** - Number of times port was closed due to a timeout • **errors** - Number of times an error occurred while opening the port • **packets in** - Number of packets received from the server • **packets out** - Number of packets sent to the server
connection	The current connection status. This can be “no connection” or “connection active”.

Authentication-method lists

To implement one or more authentication methods for securing access to the device, you configure authentication-method lists that set the order in which the authentication methods are consulted.

In an authentication-method list, you specify the access method (Telnet, SNMP, and so on) and the order in which the device tries one or more of the following authentication methods:

- Local Telnet login password
- Local password for the Super User privilege level
- Local user accounts configured on the device
- Database on a TACACS or TACACS+ server
- Database on a RADIUS server
- No authentication

NOTE
The TACACS/TACACS+, RADIUS, and Telnet login password authentication methods are not supported for SNMP access.

NOTE
To authenticate Telnet access to the CLI, you also must enable the authentication by entering the **enable telnet authentication** command at the global CONFIG level of the CLI.

NOTE
You do not need an authentication-method list to secure access based on ACLs or a list of IP addresses. Refer to "ACL usage to restrict remote access" on page 3 or "Remote access restrictions" on page 6.

In an authentication-method list for a particular access method, you can specify up to seven authentication methods. If the first authentication method is successful, the software grants access and stops the authentication process. If the access is rejected by the first authentication method, the software denies access and stops checking.

However, if an error occurs with an authentication method, the software tries the next method on the list, and so on. For example, if the first authentication method is the RADIUS server, but the link to the server is down, the software will try the next authentication method in the list.

NOTE
If an authentication method is working properly and the password (and user name, if applicable) is not known to that method, this is not an error. The authentication attempt stops, and the user is denied access.

The software will continue this process until either the authentication method is passed or the software reaches the end of the method list. If the Super User level password is not rejected after all the access methods in the list have been tried, access is granted.

Examples of authentication-method lists

The following examples show how to configure authentication-method lists. In these examples, the primary authentication method for each is "local". The device will authenticate access attempts using the locally configured usernames and passwords.

To configure an authentication-method list for SNMP, enter a command such as the following.

```
Brocade(config)# aaa authentication snmp-server default local
```

This command allows certain incoming SNMP SET operations to be authenticated using the locally configured usernames and passwords. When this command is enabled, community string validation is not performed for incoming SNMP V1 and V2c packets. This command takes effect as long as the first varbind for SNMP packets is set to one of the following:

- snAgGblPassword="*<username> <password>*" (for AAA method local)
- snAgGblPassword="*<password>*" (for AAA method line, enable)

NOTE
Certain SNMP objects need additional validation. These objects include but are not limited to: **snAgReload**, **snAgWriteNVRAM**, **snAgConfigFromNVRAM**, **snAgImgLoad**, **snAgCfgLoad** and **snAgGblTelnetPassword**. For more information, see **snAgGblPassword** in the *IronWare MIB Reference Guide.*

If AAA is set up to check both the username and password, the string contains the username, followed by a space then the password. If AAA is set up to authenticate with the current Enable or Line password, the string contains the password only.

Note that the above configuration can be overridden by the command **no snmp-server pw-check**, which disables password checking for SNMP SET requests.

Example 3

To configure an authentication-method list for the Privileged EXEC and CONFIG levels of the CLI, enter the following command.

```
Brocade(config)# aaa authentication enable default local
```

This command configures the device to use the local user accounts to authenticate attempts to access the Privileged EXEC and CONFIG levels of the CLI.

Example 4

To configure the device to consult a RADIUS server first to authenticate attempts to access the Privileged EXEC and CONFIG levels of the CLI, then consult the local user accounts if the RADIUS server is unavailable, enter the following command.

```
Brocade(config)# aaa authentication enable default radius local
```

Command Syntax

The following is the command syntax for the preceding examples.

Syntax: [**no**] **aaa authentication snmp-server** | **enable** | **login default** *method1* [*method2*] [*method3*] [*method4*] [*method5*] [*method6*] [*method7*]

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

NOTE
TACACS/TACACS+ and RADIUS are supported only with the **enable** and **login** parameters.

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

TABLE 11 Authentication method values

Method parameter	Description
line	Authenticate using the password you configured for Telnet access. The Telnet password is configured using the **enable telnet password...** command. Refer to "Setting a Telnet password" on page 13.
enable	Authenticate using the password you configured for the Super User privilege level. This password is configured using the **enable super-user-password...** command. Refer to "Setting passwords for management privilege levels" on page 14.
local	Authenticate using a local user name and password you configured on the device. Local user names and passwords are configured using the **username...** command. Refer to "Local user account configuration" on page 21.
tacacs	Authenticate using the database on a TACACS server. You also must identify the server to the device using the **tacacs-server** command.
tacacs+	Authenticate using the database on a TACACS+ server. You also must identify the server to the device using the **tacacs-server** command.
radius	Authenticate using the database on a RADIUS server. You also must identify the server to the device using the **radius-server** command. Refer to "RADIUS security" on page 41.
none	Do not use any authentication method. The device automatically permits access.

TCP Flags - edge port security

The edge port security feature works in combination with IP ACL rules, and supports all 6 TCP flags present in the offset 13 of the TCP header:

- +|- urg = Urgent
- +|- ack = Acknowledge
- +|- psh = Push
- +|- rst = Reset
- +|- syn = Synchronize
- +|- fin = Finish

TCP flags can be combined with other ACL functions (such as dscp-marking and traffic policies), giving you greater flexibility when designing ACLs.

The TCP flags feature offers two options, match-all and match-any:

- **Match-any** - Indicates that incoming TCP traffic must be matched against any of the TCP flags configured as part of the match-any ACL rule. In CAM hardware, the number of ACL rules will match the number of configured flags.
- **Match-all** - Indicates that incoming TCP traffic must be matched against all of the TCP flags configured as part of the match-all ACL rule. In CAM hardware, there will be only one ACL rule for all configured flags.

Example

```
Brocade(config-ext-nACL)# permit tcp 10.1.1.1 0.0.0.255 eq 100 10.2.2.2
0.0.0.255 eq 300 match-all +urg +ack +syn -rst
```

This command configures a single rule in CAM hardware. This rule will contain all of the configured TCP flags (urg, ack, syn, and rst).

Using TCP Flags in combination with other ACL features

The TCP Flags feature has the added capability of being combined with other ACL features.

Example

```
Brocade(config-ext-nACL)# permit tcp any any match-all +urg +ack +syn -rst
traffic-policy test
```

This command configures the ACL to match incoming traffic with the TCP Flags urg, ack, and syn and also to apply the traffic policy (rate, limit, etc.) to the matched traffic.

```
Brocade(config-ext-nACL)# permit tcp any any match-all +urg +ack +syn -rst tos
normal
```

This command configures the ACL to match incoming traffic with the flags urg, ack, and syn, and also sets the tos bit to normal when the traffic exits the device.

NOTE
TCP Flags combines the functionality of older features such as TCP Syn Attack and TCP Establish. Avoid configuring these older features on a port where you have configured TCP Flags. TCP Flags can perform all of the functions of TCP Syn Attack and TCP Establish, and more. However, if TCP Syn Attack is configured on a port along with TCP Flags, TCP Syn Attack will take precedence.

NOTE
If an ACL clause with match-any exists, and the system runs out of CAM, if the total number of TCP rules to TCP Flags will not fit within 1021 entries (the maximum rules allowed per device), then none of the TCP Flag rules will be programmed into the CAM hardware.

NOTE
If a range option and match-any TCP-flags are combined in the same ACL, the total number of rules will be calculated as: Total number of rules in CAM hardware = (number of rules for range)* (number of rules for match-any TCP-flags).

TCP Flags - edge port security

Chapter

SSH2 and SCP

2

Table 12 lists SSH2 and Secure Copy features supported on Brocade ICX 6650.

TABLE 12 Supported SSH2 and Secure Copy features

Feature	Brocade ICX 6650
Secure Shell (SSH) version 2	Yes
AES encryption for SSH2	Yes
Optional parameters for SSH2	Yes
Using secure copy (SCP) with SSH2	Yes
Filtering SSH access using ACLs	Yes
Terminating an active SSH connection	Yes
SSH client	Yes

SSH version 2 overview

Secure Shell (SSH) is a mechanism for allowing secure remote access to management functions on a Brocade device. SSH provides a function similar to Telnet. Users can log into and configure the device using a publicly or commercially available SSH client program, just as they can with Telnet. However, unlike Telnet, which provides no security, SSH provides a secure, encrypted connection to the device.

The Brocade SSH2 implementation is compatible with all versions of the SSH2 protocol (2.1, 2.2, and so on). At the beginning of an SSH session, the Brocade device negotiates the version of SSH2 to be used. The highest version of SSH2 supported by both the Brocade device and the client is the version that is used for the session. Once the SSH2 version is negotiated, the encryption algorithm with the highest security ranking is selected to be used for the session.

Brocade devices also support Secure Copy (SCP) for securely transferring files between a Brocade device and SCP-enabled remote hosts.

NOTE
The SSH feature includes software that is copyright Allegro Software Development Corporation.

SSH2 is supported in the Layer 2 and Layer 3 codes.

SSH2 is a substantial revision of Secure Shell, comprising the following hybrid protocols and definitions:

- SSH Transport Layer Protocol
- SSH Authentication Protocol
- SSH Connection Protocol
- SECSH Public Key File Format

- SSH Fingerprint Format
- SSH Protocol Assigned Numbers
- SSH Transport Layer Encryption Modes
- SCP/SSH URI Format

Tested SSH2 clients

The following SSH clients have been tested with SSH2:

- SSH Secure Shell 3.2.3
- Van Dyke SecureCRT 5.2.2
- F-Secure SSH Client 5.3 and 6.0
- PuTTY 0.60
- OpenSSH 4.3p2
- Brocade SSH Client

NOTE
Supported SSH client public key sizes are 1024 bits for DSA keys, and 1024 or 2048 bits for RSA keys.

SSH2 supported features

SSH2 (Secure Shell version 2 protocol) provides an SSH server and an SSH client. The SSH server allows secure remote access management functions on a Brocade device. SSH provides a function that is similar to Telnet, but unlike Telnet, SSH provides a secure, encrypted connection.

Brocade SSH2 support includes the following:

- Key exchange methods are **diffie-hellman-group1-sha1**
- The supported public key algorithms are **ssh-dss and ssh-rsa**.
- Encryption is provided with **3des-cbc**, **aes128-cbc**, **aes192-cbc** or **aes256-cbc**. AES encryption has been adopted by the U.S. Government as an encryption standard.
- Data integrity is ensured with **hmac-sha1**.
- Supported authentication methods are **Password** and **publickey**.
- Five inbound SSH connection at one time are supported.
- One outbound SSH is supported.

SSH2 unsupported features

The following are not supported with SSH2:

- Compression
- TCP/IP port forwarding, X11 forwarding, and secure file transfer
- SSH version 1

SSH2 authentication types

The Brocade implementation of SSH2 supports the following types of user authentication:

- *DSA challenge-response authentication*, where a collection of public keys are stored on the device. Only clients with a private key that corresponds to one of the stored public keys can gain access to the device using SSH.
- *RSA challenge-response authentication*, where a collection of public keys are stored on the device. Only clients with a private key that corresponds to one of the stored public keys can gain access to the device using SSH.
- *Password authentication*, where users attempting to gain access to the device using an SSH client are authenticated with passwords stored on the device or on a TACACS or TACACS+ server or a RADIUS server.

Configuring SSH2

You can configure the device to use any combination of these authentication types. The SSH server and client negotiate which type to use.

To configure SSH2, follow these steps:

1. Generate a host Digital Signature Algorithm (DSA) or Really Secure Algorithm (RSA) public and private key pair for the device.

   See the section "Enabling and disabling SSH by generating and deleting host keys" on page 65.

2. Configure DSA or RSA challenge-response authentication.

   See the section "Configuring DSA or RSA challenge-response authentication" on page 67.

3. Set optional parameters.

   See the section "Optional SSH parameters" on page 69.

Enabling and disabling SSH by generating and deleting host keys

To enable SSH, you generate a public and private DSA or RSA host key pair on the device. The SSH server on the Brocade device uses this host DSA or RSA key pair, along with a dynamically generated server DSA or RSA key pair, to negotiate a session key and encryption method with the client trying to connect to it.

While the SSH listener exists at all times, sessions can not be started from clients until a host key is generated. After a host key is generated, clients can start sessions.

To disable SSH, you delete all of the host keys from the device.

When a host key pair is generated, it is saved to the flash memory of all management modules. When a host key pair is is deleted, it is deleted from the flash memory of all management modules.

The time to initially generate SSH keys varies depending on the configuration, and can be from a under a minute to several minutes.

NOTE
If you have generated SSH keys on the switch, you should delete and regenerate it when you upgrade or downgrade the software version before ssh session.

Setting the CPU priority for key generation

Generating the key is a resource-intensive operation. You can set the priority for this operation to high so that the device allocates more CPU time for this operation. So you must use this option only when the device is in the maintenance window. This option reduces the time taken for key generation.

To set high priority for the key generation operation, enter the following command:

```
Brocade(config)#crypto-gen priority high
```

Syntax: crypto key crypto-gen priority default | high

The **default** keyword sets the priority as default. The key generation task is handled with the regular priority.

The **high** keyword sets the high priority for the key generation task. Use this option only when the device is in the maintenance window.

Generating and deleting a DSA key pair

To generate a DSA key pair, enter the following command.

```
Brocade(config)# crypto key generate dsa
```

To delete the DSA host key pair, enter the following command.

```
Brocade(config)# crypto key zeroize dsa
```

Syntax: crypto key generate | zeroize dsa

The **generate** keyword places a host key pair in the flash memory and enables SSH on the device, if it is not already enabled.

The **zeroize** keyword deletes the host key pair from the flash memory. This disables SSH if no other server host keys exist on the device.

The **dsa** keyword specifies a DSA host key pair. This keyword is optional. If you do not enter it, the command **crypto key generate** generates a DSA key pair by default, and the command **crypto key zeroize** works as described in "Deleting DSA and RSA key pairs" on page 67.

Generating and deleting an RSA key pair

To generate an RSA key pair, enter a command such as the following:

```
Brocade(config)# crypto key generate rsa modulus 2048
```

To delete the RSA host key pair, enter the following command.

```
Brocade(config)# crypto key zeroize rsa
```

Syntax: crypto key generate | zeroize rsa [**modulus** *modulus-size*]

The **generate** keyword places an RSA host key pair in the flash memory and enables SSH on the device, if it is not already enabled.

The optional [**modulus** *modulus-size*] parameter specifies the modulus size of the RSA key pair, in bits. The valid values for *modulus-size* are 1024 or 2048. The default value is 1024.

The **zeroize** keyword deletes the RSA host key pair from the flash memory. This disables SSH if no other authentication keys exist on the device.

The **rsa** keyword specifies an RSA host key pair.

Deleting DSA and RSA key pairs

To delete DSA and RSA key pairs from the flash memory, enter the following command:

```
Brocade(config)# crypto key zeroize
```

Syntax: crypto key zeroize

The **zeroize** keyword deletes the host key pair from the flash memory. This disables SSH.

Providing the public key to clients

The host DSA or RSA key pair is stored in the system-config file of the Brocade device. Only the public key is readable. Some SSH client programs add the public key to the known hosts file automatically. In other cases, you must manually create a known hosts file and place the public key of the Brocade device in it.

If you are using SSH to connect to a Brocade device from a UNIX system, you may need to add the public key on the Brocade device to a “known hosts” file on the client UNIX system; for example, $HOME/.ssh/known_hosts. The following is an example of an entry in a known hosts file.

```
AAAAB3NzaC1kc3MAAACBAPY8ZOHY2yFSJA6XYC9HRwNHxaehvx5wOJ0rzZdzoSOXxbET
W6ToHv8D1UJ/
z+zHo9Fiko5XybZnDIaBDHtblQ+Yp7StxyltHnXF1YLfKD1G4T6JYrdH YI14Om
1eg9e4NnCRleaqoZPF3UGfZia6bXrGTQf3gJq2e7Yisk/gF+1VAAAAFQDb8D5cv
wHWTZDPfX0D2s9Rd7NBvQAAAIEAlN92+Bb7D4KLYk3IwRbXblwXdkPggA4pfdtW9v
GfJ0/RHd+NjB4eo1D+0dix6tXwYGN7PKS5R/FXPNwxHPapcj9uL1Jn2AWQ2dsknf+i/FAA
vioUPkmdMc0zuWoSOEsSNhVDtX3WdvVcGcBq9cetzrtOKWOocJmJ80qadxTRHtUAAACB
AN7CY+KKv1gHpRzFwdQm7HK9bb1LAo2KwaoXnadFgeptNBQeSXG1vO+JsvphVMBJc9HS
n24VYtYtsMu74qXviYjziVucWKjjKEb11juqnF0GDlB3VVmxHLmxnAz643WK42Z7dLM5
sY29ouezv4Xz2PuMch5VGPP+CDqzCM4loWgV
```

Configuring DSA or RSA challenge-response authentication

With DSA or RSA challenge-response authentication, a collection of clients’ public keys are stored on the Brocade device. Clients are authenticated using these stored public keys. Only clients that have a private key that corresponds to one of the stored public keys can gain access to the device using SSH.

When DSA or RSA challenge-response authentication is enabled, the following events occur when a client attempts to gain access to the device using SSH:

1. The client sends its public key to the Brocade device.
2. The Brocade device compares the client public key to those stored in memory.
3. If there is a match, the Brocade device uses the public key to encrypt a random sequence of bytes.
4. The Brocade device sends these encrypted bytes to the client.
5. The client uses its private key to decrypt the bytes.
6. The client sends the decrypted bytes back to the Brocade device.
7. The Brocade device compares the decrypted bytes to the original bytes it sent to the client. If the two sets of bytes match, it means that the client private key corresponds to an authorized public key, and the client is authenticated.

Setting up DSA or RSA challenge-response authentication consists of the following steps.

1. Import authorized public keys into the Brocade device.
2. Enable DSA or RSA challenge response authentication.

Importing authorized public keys into the Brocade device

SSH clients that support DSA or RSA authentication normally provide a utility to generate a DSA or RSA key pair. The private key is usually stored in a password-protected file on the local host; the public key is stored in another file and is not protected. You must import the client public key for each client into the Brocade device.

Collect one public key of each key type (DSA and/or RSA) from each client to be granted access to the Brocade device and place all of these keys into one file. This public key file may contain up to 17 keys. The following is an example of a public key file containing one public key:

```
---- BEGIN SSH2 PUBLIC KEY ----
Comment: DSA Public Key
AAAAB3NzaC1kc3MAAACBAPY8ZOHY2yFSJA6XYC9HRwNHxaehvx5wOJ0rzZdzoSOXxbET
W6ToHv8D1UJ/
z+zHo9Fiko5XybZnDIaBDHtblQ+Yp7StxyltHnXF1YLfKD1G4T6JYrdH YI14Om
1eg9e4NnCRleaqoZPF3UGfZia6bXrGTQf3gJq2e7Yisk/gF+1VAAAAFQDb8D5cv
wHWTZDPfX0D2s9Rd7NBvQAAAIEAlN92+Bb7D4KLYk3IwRbXblwXdkPggA4pfdtW9v
GfJ0/RHd+NjB4eo1D+0dix6tXwYGN7PKS5R/FXPNwxHPapcj9uL1Jn2AWQ2dsknf+i/FAA
vioUPkmdMc0zuWoSOEsSNhVDtX3WdvVcGcBq9cetzrtOKWOocJmJ80qadxTRHtUAAACB
AN7CY+KKv1gHpRzFwdQm7HK9bb1LAo2KwaoXnadFgeptNBQeSXG1vO+JsvphVMBJc9HS
n24VYtYtsMu74qXviYjziVucWKjjKEb11juqnF0GDlB3VVmxHLmxnAz643WK42Z7dLM5
sY29ouezv4Xz2PuMch5VGPP+CDqzCM4loWgV
---- END SSH2 PUBLIC KEY ----
```

NOTE

Each key in the public key file must begin and end with the first and last lines in this example. If your client does not include these lines in the public key, you must manually add them.

Import the authorized public keys into the Brocade device active configuration by loading this public key file from a TFTP server.

To load a public key file called pkeys.txt from a TFTP server, enter a command such as the following:

```
Brocade(config)# ip ssh pub-key-file tftp 192.168.1.234 pkeys.txt
```

Syntax: ip ssh pub-key-file tftp *tftp-server-ip-addr filename* | **remove**

The *tftp-server-ip-addr* variable is the IP address of the tftp server that contains the public key file that you want to import into the Brocade device.

The *filename* variable is the name of the public key file that you want to import into the Brocade device.

The **remove** parameter deletes the public keys from the device.

To display the currently loaded public keys, enter the following command.

```
Brocade# show ip client-pub-key
---- BEGIN SSH2 PUBLIC KEY ----
Comment: DSA Public Key
AAAAB3NzaC1kc3MAAACBAPY8ZOHY2yFSJA6XYC9HRwNHxaehvx5wOJ0rzZdzoSOXxbET
W6ToHv8D1UJ/
z+zHo9Fiko5XybZnDIaBDHtblQ+Yp7StxyltHnXF1YLfKD1G4T6JYrdH YI14Om
1eg9e4NnCRleaqoZPF3UGfZia6bXrGTQf3gJq2e7Yisk/gF+1VAAAAFQDb8D5cv
wHWTZDPfX0D2s9Rd7NBvQAAAIEAlN92+Bb7D4KLYk3IwRbXblwXdkPggA4pfdtW9v
GfJ0/RHd+NjB4eo1D+0dix6tXwYGN7PKS5R/FXPNwxHPapcj9uL1Jn2AWQ2dsknf+i/FAA
vioUPkmdMc0zuWoSOEsSNhVDtX3WdvVcGcBq9cetzrtOKWOocJmJ80qadxTRHtUAAACB
AN7CY+KKv1gHpRzFwdQm7HK9bb1LAo2KwaoXnadFgeptNBQeSXG1vO+JsvphVMBJc9HS
n24VYtYtsMu74qXviYjziVucWKjjKEb11juqnF0GDlB3VVmxHLmxnAz643WK42Z7dLM5
sY29ouezv4Xz2PuMch5VGPP+CDqzCM4loWgV
---- END SSH2 PUBLIC KEY ----
```

Syntax: show ip client-pub-key [**begin** *expression* | **exclude** *expression* | **include** *expression*]

To clear the public keys from the buffers, enter the following command.

```
Brocade# clear public-key
```

Syntax: clear public-key

Enabling DSA or RSA challenge-response authentication

DSA and RSA challenge-response authentication is enabled by default. You can disable or re-enable it manually.

To enable DSA and RSA challenge-response authentication.

```
Brocade(config)# ip ssh key-authentication yes
```

To disable DSA and RSA challenge-response authentication.

```
Brocade(config)# ip ssh key-authentication no
```

Syntax: ip ssh key-authentication yes | **no**

Optional SSH parameters

You can adjust the following SSH settings on the Brocade device:

- The number of SSH authentication retries
- The user authentication method the Brocade device uses for SSH connections

- Whether the Brocade device allows users to log in without supplying a password
- The port number for SSH connections
- The SSH login timeout value
- A specific interface to be used as the source for all SSH traffic from the device
- The maximum idle time for SSH sessions

Setting the number of SSH authentication retries

By default, the Brocade device attempts to negotiate a connection with the connecting host three times. The number of authentication retries can be changed to between 1–5.

For example, the following command changes the number of authentication retries to 5.

```
Brocade(config)# ip ssh authentication-retries 5
```

Syntax: ip ssh authentication-retries *number*

Deactivating user authentication

After the SSH server on the Brocade device negotiates a session key and encryption method with the connecting client, user authentication takes place. The Brocade implementation of SSH supports DSA or RSA challenge-response authentication and password authentication.

With DSA or RSA challenge-response authentication, a collection of clients' public keys are stored on the Brocade device. Clients are authenticated using these stored public keys. Only clients that have a private key that corresponds to one of the stored public keys can gain access to the device using SSH.

With password authentication, users are prompted for a password when they attempt to log into the device (provided empty password logins are not allowed). If there is no user account that matches the user name and password supplied by the user, the user is not granted access.

You can deactivate one or both user authentication methods for SSH. Note that deactivating both authentication methods essentially disables the SSH server entirely.

To disable DSA or RSA challenge-response authentication, enter the following command.

```
Brocade(config)# ip ssh key-authentication no
```

Syntax: ip ssh key-authentication yes | no

The default is **yes**.

To deactivate password authentication, enter the following command.

```
Brocade(config)# ip ssh password-authentication no
```

Syntax: ip ssh password-authentication no | yes

The default is **yes**.

Enabling empty password logins

By default, empty password logins are not allowed. This means that users with an SSH client are always prompted for a password when they log into the device. To gain access to the device, each user must have a user name and password. Without a user name and password, a user is not granted access.

If you enable empty password logins, users are ***not*** prompted for a password when they log in. Any user with an SSH client can log in without being prompted for a password.

To enable empty password logins, enter the following command.

```
Brocade(config)# ip ssh permit-empty-passwd yes
```

Syntax: ip ssh permit-empty-passwd no | yes

Setting the SSH port number

By default, SSH traffic occurs on TCP port 22. You can change this port number. For example, the following command changes the SSH port number to 2200.

```
Brocade(config)# ip ssh port 2200
```

Note that if you change the default SSH port number, you must configure SSH clients to connect to the new port. Also, you should be careful not to assign SSH to a port that is used by another service. If you change the SSH port number, Brocade recommends that you change it to a port number greater than 1024.

Syntax: ip ssh port *number*

Setting the SSH login timeout value

When the SSH server attempts to negotiate a session key and encryption method with a connecting client, it waits a maximum of 120 seconds for a response from the client. If there is no response from the client after 120 seconds, the SSH server disconnects. You can change this timeout value to between 1–120 seconds. For example, to change the timeout value to 60 seconds, enter the following command.

```
Brocade(config)# ip ssh timeout 60
```

Syntax: ip ssh timeout *seconds*

Designating an interface as the source for all SSH packets

You can designate a loopback interface, virtual interface, or Ethernet port as the source for all SSH packets from the device. For more information, refer to *Brocade ICX 6650 Layer 3 Routing Configuration Guide*.

Configuring the maximum idle time for SSH sessions

By default, SSH sessions do not time out. Optionally, you can set the amount of time an SSH session can be inactive before the Brocade device closes it. For example, to set the maximum idle time for SSH sessions to 30 minutes, enter the following command.

```
Brocade(config)# ip ssh idle-time 30
```

Syntax: ip ssh idle-time *minutes*

If an established SSH session has no activity for the specified number of minutes, the Brocade device closes it. An idle time of 0 minutes (the default value) means that SSH sessions never time out. The maximum idle time for SSH sessions is 240 minutes.

Filtering SSH access using ACLs

You can permit or deny SSH access to the Brocade device using ACLs. To use ACLs, first create the ACLs you want to use. You can specify a numbered standard IPv4 ACL, a named standard IPv4 ACL

Enter commands such as the following.

```
Brocade(config)# access-list 10 permit host 192.168.144.241
Brocade(config)# access-list 10 deny host 192.168.144.242 log
Brocade(config)# access-list 10 permit host 192.168.144.243
Brocade(config)# access-list 10 deny any
Brocade(config)# ssh access-group 10
```

Syntax: ssh access-group *standard-named-acl* | *standard-numbered-acl*

Terminating an active SSH connection

To terminate one of the active SSH connections, enter the following command

```
Brocade# kill ssh 1
```

Syntax: kill ssh *connection-id*

Displaying SSH information

Up to five SSH connections can be active on the Brocade device.

Displaying SSH connection information

To display information about SSH connections, enter the **show ip ssh** command.

```
Brocade# show ip ssh
Connection    Version    Encryption    Username    HMAC       Server Hostkey       IP Address
Inbound:
     1         SSH-2      3des-cbc     Raymond     hmac-sha1  ssh-dss              10.120.54.2
Outbound:
     6         SSH-2      aes256-cbc   Steve       hmac-sha1  ssh-dss              10.37.77.15

SSH-v2.0 enabled; hostkey: DSA(1024), RSA(2048)
```

Syntax: show ip ssh [**begin** *expression* | **exclude** *expression* | **include** *expression*]

This display shows the following information about the active SSH connections.

TABLE 13 SSH connection information

Field	Description
Inbound	Connections listed under this heading are inbound.
Outbound	Connections listed under this heading are outbound.
Connection	The SSH connection ID.
Version	The SSH version number.
Encryption	The encryption method used for the connection.
Username	The user name for the connection.
HMAC	The HMAC version
Server Hostkey	The type of server hostkey. This can be DSA or RSA.
IP Address	The IP address of the SSH client
SSH-v2.0 enabled	Indicates that SSHv2 is enabled.
hostkey	Indicates that at least one host key is on the device. It is followed by a list of the host key types and modulus sizes.

Displaying SSH configuration information

To display SSH configuration information, use the **show ip ssh config** command:

```
Brocade# show ip ssh config
SSH server           :Enabled
SSH port             :22
Encryption           :AES-256 AES-192 AES-128 3-DES
Permit empty password       :Yes
Authentication methods      :Password Public-key Interactive
Authentication retries   :10
Login timeout (seconds)     :20
Idle timeout (minutes)   :10
Strict management VRF       :Enabled
SCP                  :Disabled
SSH IPv4 clients         :10.200.200.201. 10.200.200.202. 10.200.200.203
SSH IPv6 clients         :2001:DB8:4545:3112:2040:f8ff:fe21:6001
SSH IPv4 access-list     :4
SSH IPv6 access-list     :ssh_ipv6_acl
Brocade#
```

Syntax: show ip ssh config

This display shows the following information.

TABLE 14 SSH configuration information

Field	Description
SSH server	SSH server is enabled or disabled
SSH port	SSH port number

TABLE 14 SSH configuration information (Continued)

Field	Description
Encryption	The encryption used for the SSH connection. The following values are displayed when AES only is enabled: • AES-256, AES-192, and AES-128 indicate the different AES methods used for encryption. • 3-DES indicates 3-DES algorithm is used for encryption.
Permit empty password	Empty password login is allowed or not allowed.
Authentication methods	The authentication methods used for SSH. The authentication can have one or more of the following values: • **Password** - indicates that you are prompted for a password when attempting to log into the device. • **Public-key** - indicates that DSA or RSA challenge-response authentication is enabled. • **Interactive** - indicates the interactive authentication si enabled.
Authentication retries	The number of authentication retries. This number can be from 1 to 5.
Login timeout (seconds)	SSH login timeout value in seconds. This can be from 0 to 120.
Idle timeout (minutes)	SSH idle timeout value in minutes. This can be from 0 to 240.
Strict management VRF	Strict management VRF is enabled or disabled.
SCP	SCP is enabled or disabled.
SSH IPv4 clients	The list of IPv4 addresses to which SSH access is allowed. The default is “All”.
SSH IPv6 clients	The list of IPv4 addresses to which SSh access is allowed. Default “All”.
SSH IPv4 access-list	The IPv4 ACL used to permit or deny access using SSH.
SSH IPv6 access-list	The IPv6 ACL used to permit or deny access to device using SSH.

Displaying additional SSH connection information

The **show who** command also displays information about SSH connections:

```
Brocade# show who
       Console connections:
       Established
       you are connecting to this session
       2 minutes 56 seconds in idle

SSH server status: Enabled
SSH connections (inbound):
1. established, client ip address 10.2.2.1, server hostkey DSA
   1 minutes 15 seconds in idle
2. established, client ip address 10.2.2.2, server hostkey RSA
   2 minutes 25 seconds in idle
SSH connection (outbound):
3. established, server ip address 10.37.77.15, server hostkey RSA
    7 seconds in idle
```

show who [**begin** *expression* | **exclude** *expression* | **include** *expression*]

Secure copy with SSH2

Secure Copy (SCP) uses security built into SSH to transfer image and configuration files to and from the device. SCP automatically uses the authentication methods, encryption algorithm, and data compression level configured for SSH. For example, if password authentication is enabled for SSH, the user is prompted for a user name and password before SCP allows a file to be transferred. No additional configuration is required for SCP on top of SSH.

You can use SCP to copy files on the Brocade device, including the startup configuration and running configuration files, to or from an SCP-enabled remote host.

Enabling and disabling SCP

SCP is enabled by default and can be disabled. To disable SCP, enter the following command.

```
Brocade(config)# ip ssh scp disable
```

Syntax: ip ssh scp disable | enable

NOTE
If you disable SSH, SCP is also disabled.

Secure copy configuration notes

- When using SCP, enter the **scp** commands on the SCP-enabled client, rather than the console on the Brocade device.
- Certain SCP client options, including -p and -r, are ignored by the SCP server on the Brocade device. If an option is ignored, the client is notified.
- An SCP AES copy of the running or start configuration file from the Brocade device to Linux WS 4 or 5 may fail if the configuration size is less than 700 bytes. To work around this issue, use PuTTY to copy the file.

Example file transfers using SCP

The following are examples of using SCP to transfer files to and from a Brocade device.

Copying a file to the running configuration

To copy a configuration file (c:\cfg\brocade.cfg) to the running configuration file on a Brocade device at 192.168.1.50 and log in as user terry, enter the following command on the SCP-enabled client.

```
C:\> scp c:\cfg\brocade.cfg terry@192.168.1.50:runConfig
```

If password authentication is enabled for SSH, the user is prompted for user terry password before the file transfer takes place.

Copying a file to the startup configuration

To copy the configuration file to the startup configuration file, enter the following command.

```
C:\> scp c:\cfg\brocade.cfg terry@192.168.1.50:startConfig
```

Copying the running config uration file to an SCP-enabled client

To copy the running configuration file on the Brocade device to a file called c:\cfg\fdryrun.cfg on the SCP-enabled client, enter the following command.

```
C:\> scp terry@192.168.1.50:runConfig c:\cfg\brcdrun.cfg
```

Copying the startup configuration file to an SCP-enabled client

To copy the startup configuration file on the Brocade device to a file called c:\cfg\brcdestart.cfg on the SCP-enabled client, enter the following command.

```
C:\> scp terry@192.168.1.50:startConfig c:\cfg\brcdstart.cfg
```

To overwrite the running configuration file

```
C:\> scp c:\cfg\brocade.cfg terry@192.168.1.50:runConfig-overwrite
```

Copying a software image file to flash memory

To copy a software image file from an SCP-enabled client to the **primary** flash on these devices, enter one of the following commands.

```
C:\> scp FCXLR07500.bin terry@192.168.1.50:flash:primary
```
or
```
C:\> scp terry@192.168.1.50:flash:primary FCXLR07500.bin
```

To copy a software image file from an SCP-enabled client to the **secondary** flash on these devices, enter one of the following commands.

```
C:\> scp FCXLR07500.bin terry@192.168.1.50:flash:secondary
```
or
```
C:\> scp terry@192.168.1.50:flash:secondary FCXLR07500.bin
```

NOTE
The Brocade device supports only one SCP copy session at a time.

Copying a software image file from flash memory

To copy a software image file from the **primary** flash on these devices to an SCP-enabled client, enter a command such as the following.

```
C:\> scp terry@192.168.1.50:flash:primary FCXLR07500.bin
```

To copy a software image file from the **secondary** flash on these devices to an SCP-enabled client, enter a command such as the following.

```
C:\> scp terry@192.168.1.50:flash:secondary FCXLR07500.bin
```

NOTE
The Brocade device supports only one SCP copy session at a time.

Importing a digital certificate using SCP

To import a digital certificate using SCP, enter a command such as the following one:

```
C:\> scp certfile user@192.168.89.210:sslCert
```

Syntax: scp *certificate-filename user@ip-address*:**sslCert.**

The *ip-address* variable is the IP address of the server from which the digital certificate file is downloaded.

The *certificate-filename* variable is the file name of the digital certificate that you are importing to the device.

The **scp** command can be used when TFTP access is unavailable or not permitted and the command has an equivalent functionality to the **ip ssl certificate-data-file tftp**. For more information on the **ip ssl certificate-data-file tftp** command, refer to "Importing digital certificates and RSA private key files" on page 27.

Importing an RSA private key

To import an RSA private key from a client using SCP, enter a command such as the following one:

```
C:\> scp keyfile user@192.168.9.210:sslPrivKey
```

Syntax: scp *key-filename user@ip-address:* **sslPrivKey**

The *ip-address* variable is the IP address of the server that contains the private key file.

The *key-filename* variable is the file name of the private key that you want to import into the device.

The **scp** command can be used when TFTP access is unavailable or not permitted and the command has an equivalent functionality to the **ip ssl private-key-file tftp** command. For more information on the **ip ssl private-key-file tftp** command, refer to "Importing digital certificates and RSA private key files" on page 27.

Importing a DSA or RSA public key

To import a DSA or RSA public key from a client using SCP, enter a command such as the following one:

```
C:\> scp pkeys.txt user@192.168.1.234:sshPubKey
```

Syntax: scp *key-filename user@ip-address*:**sshPubKey**

The *ip-address* variable is the IP address of the server that contains the public key file.

The *key-filename* variable is the name of the DSA or RSA public key file that you want to import into the device.

The **scp** command can be used when TFTP access is unavailable or not permitted and the command has an equivalent function to the **ip ssh pub-key-file tftp** command. For more information on the **ip ssh pub-key-file tftp** command, refer to "Importing authorized public keys into the Brocade device" on page 68.

SSH2 client

SSH2 client allows you to connect from a Brocade device to an SSH2 server, including another Brocade device that is configured as an SSH2 server. You can start an outbound SSH2 client session while you are connected to the device by any connection method (SSH2, Telnet, console). Brocade devices support one outbound SSH2 client session at a time.

The supported SSH2 client features are as follows:

- Encryption algorithms, in the order of preference:
  - aes256-cbc
  - aes192-cbc
  - aes128-cbc
  - 3des-cbc
- SSH2 client session authentication algorithms:
  - Password authentication
  - Public Key authentication
- Message Authentication Code (MAC) algorithm: hmac-sha1
- Key exchange algorithm: diffie-hellman-group1-sha1
- No compression algorithms are supported.
- The client session can be established through either in-band or out-of-band management ports.
- The client session can be established through IPv4 or IPv6 protocol access.
- The client session can be established to a server listening on a non-default SSH port.

Enabling SSH2 client

To use SSH2 client, you must first enable SSH2 server on the device. See "SSH2 authentication types" on page 65.

When SSH2 server is enabled, you can use SSH client to connect to an SSH server using password authentication.

Configuring SSH2 client public key authentication

To use SSH client for public key authentication, you must generate SSH client authentication keys and export the public key to the SSH servers to which you want to connect.

The following sections describe how to configure SSH client public key authentication:

- "Generating and deleting a client DSA key pair" on page 79
- "Generating and deleting a client RSA key pair" on page 79

- "Exporting client public keys" on page 79

Generating and deleting a client DSA key pair

To generate a client DSA key pair, enter the following command.

```
Brocade(config)# crypto key client generate dsa
```

To delete the DSA host key pair, enter the following command.

```
Brocade(config)# crypto key client zeroize dsa
```

Syntax: crypto key client generate | zeroize dsa

The **generate** keyword places a host key pair in the flash memory.

The **zeroize** keyword deletes the host key pair from the flash memory.

The **dsa** keyword specifies a DSA host key pair.

Generating and deleting a client RSA key pair

To generate a client RSA key pair, enter a command such as the following:

```
Brocade(config)# crypto key client generate rsa modulus 2048
```

To delete the RSA host key pair, enter the following command.

```
Brocade(config)# crypto key client zeroize rsa
```

Syntax: crypto key client generate | zeroize rsa [**modulus** *modulus-size*]

The **generate** keyword places an RSA host key pair in the flash memory.

The **zeroize** keyword deletes the RSA host key pair from the flash memory.

The optional [**modulus** *modulus-size*] parameter specifies the modulus size of the RSA key pair, in bits. The valid values for *modulus-size* are 1024 or 2048. It is used only with the **generate** parameter. The default value is 1024.

The **rsa** keyword specifies an RSA host key pair.

Exporting client public keys

Client public keys are stored in the following files in flash memory:

- A DSA key is stored in the file **$$sshdsapub.key**.
- An RSA key is stored in the file **$$sshrsapub.key**.

To copy key files to a TFTP server, you can use the **copy flash tftp** command.

You must copy the public key to the SSH server. If the SSH server is a brocade device, see the section "Importing authorized public keys into the Brocade device" on page 68.

Using SSH2 client

To start an SSH2 client connection to an SSH2 server using password authentication, enter a command such as the following:

```
Brocade# ssh 10.10.10.2
```

To start an SSH2 client connection to an SSH2 server using public key authentication, enter a command such as the following:

```
Brocade# ssh 10.10.10.2 public-key dsa
```

Syntax: ssh *ipv4Addr* | *ipv6Addr* | *host-name* [**public-key** [**dsa** | **rsa**]] [**port** *portnum*]

The *ipv4Addr* | *ipv6Addr* | *host-name* variable identifies an SSH2 server. You identify the server to connect to by entering its IPv4 or IPv6 address or its hostname.

The optional [**public-key** [**dsa** | **rsa**]] parameter specifies the type of public key authentication to use for the connection, either DSA or RSA. If you do not enter this parameter, the default authentication type is password.

The optional **port** *portnum* parameter specifies that the SSH2 connection will use a non-default SSH2 port, where *portnum* is the port number. The default port number is 22.

Displaying SSH2 client information

For information about displaying SSH2 client information, see the following sections:

- "Displaying SSH connection information" on page 72
- "Displaying additional SSH connection information" on page 74

Chapter

3

Rule-Based IP ACLs

Table 15 and Table 16 list the Access Control List (ACL) features supported on Brocade ICX 6650. Table 15 lists the features supported on inbound traffic, while Table 16 lists the features supported on outbound traffic. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 15 Supported ACL features on inbound traffic

Feature	Brocade ICX 6650
Hardware-based ACLs	Yes
Standard named and numbered ACLs	Yes
Extended named and numbered ACLs	Yes
User input preservation for ACL TCP/UDP port numbers	Yes
ACL comment text	Yes
ACL logging of denied packets	Yes
ACL logging with traffic rate limiting (to prevent CPU overload)	Yes **NOTE:** This feature is enabled by default. There is no CLI command to enable or disable it
Strict control of ACL filtering of fragmented packets	Yes
ACL support for switched traffic in the router image	Yes **NOTE:** This feature is enabled by default. There is no CLI command to enable or disable it
ACL filtering based on VLAN membership or VE port membership	Yes
Filtering on IP precedence and ToS value	Yes
QoS options for IP ACLs	Yes
Priority mapping using ACLs	Yes
Hardware usage statistics	Yes
Policy-based routing (PBR) (Supported in the full Layer 3 code only)	Yes

TABLE 16 Supported ACL features on outbound traffic

Feature	Brocade ICX 6650
Hardware-based ACLs	Yes
Standard named and numbered ACLs	Yes

TABLE 16 Supported ACL features on outbound traffic (Continued)

Feature	Brocade ICX 6650
Extended named and numbered ACLs	Yes
User input preservation for ACL TCP/UDP port numbers	Yes
ACL comment text	Yes
Strict control of ACL filtering of fragmented packets	Yes
ACL support for switched traffic in the router image	Yes **NOTE:** This feature is enabled by default for outbound ACLs on platforms that support outbound ACL support. There is no CLI command to enable or disable it.
Filtering on IP precedence and ToS value	Yes
QoS options for IP ACLs	Yes
Hardware usage statistics	Yes

This chapter describes how Access Control Lists (ACLs) are implemented and configured in the Brocade devices.

NOTE
For information about IPv6 ACLs, refer to Chapter 4, "IPv6 ACLs".

ACL overview

Brocade devices support rule-based ACLs (sometimes called hardware-based ACLs), where the decisions to permit or deny packets are processed in hardware and all permitted packets are switched or routed in hardware. All denied packets are also dropped in hardware. Brocade ICX 6650 support both inbound and outbound ACLs. The ACL features supported on inbound and outbound traffic are as listed in Table 15 and Table 16 respectively and discussed in more detail in the rest of this chapter.

Brocade ICX 6650 devices do not support flow-based ACLs.

Rule-based ACLs program the ACL entries you assign to an interface into Content Addressable Memory (CAM) space allocated for the ports. The ACLs are programmed into hardware at startup (or as new ACLs are entered and bound to ports). Devices that use rule-based ACLs program the ACLs into the CAM entries and use these entries to permit or deny packets in the hardware, without sending the packets to the CPU for processing.

Rule-based ACLs are supported on the following interface types:

- Gbps Ethernet ports
- 10 Gbps Ethernet ports
- Trunk groups

- Virtual routing interfaces

Types of IP ACLs

You can configure the following types of IP ACLs:

- ***Standard*** – Permits or denies packets based on source IP address. Valid standard ACL IDs are 1 – 99 or a character string.
- ***Extended*** – Permits or denies packets based on source and destination IP address and also based on IP protocol information. Valid extended ACL IDs are a number from 100 – 199 or a character string.

ACL IDs and entries

ACLs consist of ACL IDs and ACL entries:

- ***ACL ID*** – An ACL ID is a number from 1 – 99 (for a standard ACL) or 100 – 199 (for an extended ACL) or a character string. The ACL ID identifies a collection of individual ACL entries. When you apply ACL entries to an interface, you do so by applying the ACL ID that contains the ACL entries to the interface, instead of applying the individual entries to the interface. This makes applying large groups of access filters (ACL entries) to interfaces simple. Refer to "Numbered and named ACLs" on page 83.

  NOTE
  This is different from IP access policies. If you use IP access policies, you apply the individual policies to interfaces.

- ***ACL entry*** – Also called an ***ACL rule***, this is a filter command associated with an ACL ID. The maximum number of ACL rules you can configure is a system-wide parameter and depends on the device you are configuring. You can configure up to the maximum number of entries in any combination in different ACLs. For Brocade ICX 6650, the maximum number of ACL TCAM entries per port region are 2045 and maximum number of ACL entries per system is 8192. You configure ACLs on a global basis, then apply them to the incoming or outgoing traffic on specific ports. The software applies the entries within an ACL in the order they appear in the ACL configuration. As soon as a match is found, the software takes the action specified in the ACL entry (permit or deny the packet) and stops further comparison for that packet.

Numbered and named ACLs

When you configure an ACL, you can refer to the ACL by a numeric ID or by an alphanumeric name. The commands to configure numbered ACLs are different from the commands for named ACLs.

- ***Numbered ACL*** – If you refer to the ACL by a numeric ID, you can use 1 – 99 for a standard ACL or 100 – 199 for an extended ACL.
- ***Named ACL*** – If you refer to the ACL by a name, you specify whether the ACL is a standard ACL or an extended ACL, then specify the name.

You can configure up to 99 standard numbered IP ACLs and 100 extended numbered IP ACLs. You also can configure up to 99 standard named ACLs and 100 extended named ACLs.

Default ACL action

The default action when no ACLs are configured on a device is to permit all traffic. However, after you configure an ACL and apply it to a port, the default action for that port is to deny all traffic that is not explicitly permitted on the port:

- If you want to tightly control access, configure ACLs consisting of permit entries for the access you want to permit. The ACLs implicitly deny all other access.
- If you want to secure access in environments with many users, you might want to configure ACLs that consist of explicit deny entries, then add an entry to permit all access to the end of each ACL. The software permits packets that are not denied by the deny entries.

How hardware-based ACLs work

When you bind an ACL to inbound or outbound traffic on an interface, the device programs the Layer 4 CAM with the ACL. Permit and deny rules are programmed. Most ACL rules require one Layer 4 CAM entry. However, ACL rules that match on more than one TCP or UDP application port may require several CAM entries. The Layer 4 CAM entries for ACLs do not age out. They remain in the CAM until you remove the ACL:

- If a packet received on the interface matches an ACL rule in the Layer 4 CAM, the device permits or denies the packet according to the ACL.
- If a packet does not match an ACL rule, the packet is dropped, since the default action on an interface that has ACLs is to deny the packet.

How fragmented packets are processed

The descriptions above apply to non-fragmented packets. The default processing of fragments by hardware-based ACLs is as follows:

- The first fragment of a packet is permitted or denied using the ACLs. The first fragment is handled the same way as non-fragmented packets, since the first fragment contains the Layer 4 source and destination application port numbers. The device uses the Layer 4 CAM entry if one is programmed, or applies the interface's ACL entries to the packet and permits or denies the packet according to the first matching ACL.
- For other fragments of the same packet, they are subject to a rule only if there is no Layer 4 information in the rule or in any preceding rules.

The fragments are forwarded even if the first fragment, which contains the Layer 4 information, was denied. Generally, denying the first fragment of a packet is sufficient, since a transaction cannot be completed without the entire packet.

For tighter control, you can configure the port to drop all packet fragments. Refer to "Enabling strict control of ACL filtering of fragmented packets" on page 108.

Hardware aging of Layer 4 CAM entries

Rule-based ACLs use Layer 4 CAM entries. The device permanently programs rule-based ACLs into the CAM. The entries never age out.

ACL configuration considerations

- See "ACL overview" on page 82 for details on which devices support inbound and outbound ACLs.
- Hardware-based ACLs are supported on the following devices:
  - Gbps Ethernet ports
  - 10 Gbps Ethernet ports
  - Trunk groups
  - Virtual routing interfaces
- Inbound ACLs apply to all traffic, including management traffic. By default outbound ACLs are not applied to traffic generated by the CPU. This must be enabled using the enable egress-acl-on-control-traffic command. See "Applying egress ACLs to Control (CPU) traffic" on page 101 for details.
- Hardware-based ACLs support only one ACL per port. The ACL of course can contain multiple entries (rules). For example, hardware-based ACLs do not support ACLs 101 and 102 on port 1, but hardware-based ACLs do support ACL 101 containing multiple entries.
- For devices that support both, inbound ACLs and outbound ACLs can co-exist. When an inbound ACL and an outbound ACL are configured on the same port, the outbound ACL is applied only on outgoing traffic.
- ACLs are affected by port regions. Each ACL group must contain one entry for the implicit *deny all IP traffic* clause. Also, each ACL group uses a multiple of 8 ACL entries. For example, if all ACL groups contain 5 ACL entries, you could add 127ACL groups (1016/8) in that port region. If all your ACL groups contain 8 ACL entries, you could add 63 ACL groups, since you must account for the implicit deny entry.
- By default, the first fragment of a fragmented packet received by the Brocade device is permitted or denied using the ACLs, but subsequent fragments of the same packet are forwarded in hardware. Generally, denying the first fragment of a packet is sufficient, since a transaction cannot be completed without the entire packet.
- ACLs are supported on member ports of a VLAN on which DHCP snooping and Dynamic ARP Inspection (DAI) are enabled. Also, IP source guard and ACLs are supported together on the same port, as long as both features are configured at the port-level or per-port-per-VLAN level. Brocade ports do not support IP source guard and ACLs on the same port if one is configured at the port-level and the other is configured at the per-port-per-VLAN level.
- Ingress MAC filters can be applied to the same port as an outbound ACL.
- A DOS attack configuration on a port will only apply on the ingress traffic.
- Outbound ACLs cannot be configured through a RADIUS server as dynamic or user-based ACLs. However, outbound ACLs can still be configured with MAC-AUTH/DOT1X enabled, as they the two are configured in different directions.
- The following ACL features and options are not supported on the Brocade ICX 6650 devices:
  - Applying an ACL on a device that has Super Aggregated VLANs (SAVs) enabled.
  - ACL logging of permitted packets– ACL logging is supported for packets that are sent to the CPU for processing (denied packets) for inbound traffic. ACL logging is not supported for packets that are processed in hardware (permitted packets).
  - Flow-based ACLs
  - Layer 2 ACLs

- You can apply an ACL to a port that has TCP SYN protection or ICMP smurf protection, or both, enabled.

Configuring standard numbered ACLs

This section describes how to configure standard numbered ACLs with numeric IDs and provides configuration examples.

Standard ACLs permit or deny packets based on source IP address. You can configure up to 99 standard numbered ACLs. There is no limit to the number of ACL entries an ACL can contain except for the system-wide limitation. For the number of ACL entries supported on a device, refer to "ACL IDs and entries" on page 83.

Standard numbered ACL syntax

Syntax: [**no**] **access-list** *ACL-num* **deny** | **permit** *source-ip* | *hostname wildcard* [**log**]

or

Syntax: [**no**] **access-list** *ACL-num* **deny** | **permit** *source-ip/mask-bits* | *hostname* [**log**]

Syntax: [**no**] **access-list** *ACL-num* **deny** | **permit host** *source-ip* | *hostname* [**log**]

Syntax: [**no**] **access-list** *ACL-num* **deny** | **permit any** [**log**]

Syntax: [**no**] **ip access-group** *ACL-num* **in** | **out**

The *ACL-num* parameter is the access list number from 1–99.

The **deny** | **permit** parameter indicates whether packets that match a policy in the access list are denied (dropped) or permitted (forwarded).

The *source-ip* parameter specifies the source IP address. Alternatively, you can specify the host name.

NOTE
To specify the host name instead of the IP address, the host name must be configured using the DNS resolver on the Brocade device. To configure the DNS resolver name, use the **ip dns server-address**... command at the global CONFIG level of the CLI.

The *wildcard* parameter specifies the mask value to compare against the host address specified by the *source-ip* parameter. The *wildcard* is in dotted-decimal notation (IP address format). It is a four-part value, where each part is 8 bits (one byte) separated by dots, and each bit is a one or a zero. Each part is a number ranging from 0 to 255, for example 0.0.0.255. Zeros in the mask mean the packet source address must match the *source-ip.* Ones mean any value matches. For example, the *source-ip* and *wildcard* values 10.157.22.26 0.0.0.255 mean that all hosts in the Class A subnet 10.157.22.x match the policy.

If you prefer to specify the wildcard (mask value) in CIDR format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of "10.157.22.26 0.0.0.255" as "10.157.22.26/24". The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the

significant bits) and changes the non-significant portion of the IP address into ones. For example, if you specify 10.157.22.26/24 or 10.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 10.157.22.0/24 (if you have enabled display of subnet lengths) or 10.157.22.0 0.0.0.255 in the startup-config file.

If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in "*/mask-bits*" format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

NOTE
If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show ip access-list** command.

The **host** *source-ip | hostname* parameter lets you specify a host IP address or name. When you use this parameter, you do not need to specify the mask. A mask of all zeros (0.0.0.0) is implied.

The **any** parameter configures the policy to match on all host addresses.

The **log** argument configures the device to generate syslog entries and SNMP traps for inbound packets that are denied by the access policy.

The **in | out** parameter applies the ACL to incoming or outgoing traffic on the interface to which you apply the ACL. You can apply the ACL to an Ethernet port, or virtual interface.

NOTE
If the ACL is for a virtual routing interface, you also can specify a subset of ports within the VLAN containing that interface when assigning an ACL to the interface.

Configuration example for standard numbered ACLs

To configure a standard ACL and apply it to incoming traffic on port 1/1/1, enter the following commands.

```
Brocade(config)# access-list 1 deny host 10.157.22.26 log
Brocade(config)# access-list 1 deny 10.157.29.12 log
Brocade(config)# access-list 1 deny host IPHost1 log
Brocade(config)# access-list 1 permit any
Brocade(config)# int eth 1/1/1
Brocade(config-if-e10000-1/1/1)# ip access-group 1 in
Brocade(config)# write memory
```

The commands in this example configure an ACL to deny packets from three source IP addresses from being received on port 1/1/1. The last ACL entry in this ACL permits all packets that are not explicitly denied by the first three ACL entries.

Standard named ACL configuration

This section describes how to configure standard named ACLs with alphanumeric IDs. This section also provides configuration examples.

Standard ACLs permit or deny packets based on source IP address. You can configure up to 99 standard named ACLs. There is no limit to the number of ACL entries an ACL can contain except for the system-wide limitation. For the number of ACL entries supported on a device, refer to "ACL IDs and entries" on page 83.

The commands for configuring named ACL entries are different from the commands for configuring numbered ACL entries. The command to configure a numbered ACL is **access-list**. The command for configuring a named ACL is **ip access-list**. In addition, when you configure a numbered ACL entry, you specify all the command parameters on the same command. When you configure a named ACL, you specify the ACL type (standard or extended) and the ACL name with one command, which places you in the configuration level for that ACL. Once you enter the configuration level for the ACL, the command syntax is the same as the syntax for numbered ACLs.

Standard named ACL syntax

Syntax: [**no**] **ip access-list standard** *ACL-name* | *ACL-num*

Syntax: **deny** | **permit** *source-ip* | *hostname wildcard* [**log**]

or

Syntax: **deny** | **permit** *source-ip/mask-bits* | *hostname* [**log**]

Syntax: **deny** | **permit host** *source-ip* | *hostname* [**log**]

Syntax: **deny** | **permit any** [**log**]

Syntax: [**no**] **ip access-group** *ACL-name* **in** | **out**

The *ACL-name* parameter is the access list name. You can specify a string of up to 256 alphanumeric characters. You can use blanks in the ACL name if you enclose the name in quotation marks (for example, "ACL for Net1").

The *ACL-num* parameter allows you to specify an ACL number if you prefer. If you specify a number, you can specify from 1–99 for standard ACLs.

NOTE
For convenience, the software allows you to configure numbered ACLs using the syntax for named ACLs. The software also still supports the older syntax for numbered ACLs. Although the software allows both methods for configuring numbered ACLs, numbered ACLs are always formatted in the startup-config and running-config files in using the older syntax, as follows.

```
access-list 1 deny host 10.157.22.26 log
access-list 1 deny 10.157.22.0 0.0.0.255 log
access-list 1 permit any
access-list 101 deny tcp any any eq http log
```

The **deny** | **permit** parameter indicates whether packets that match a policy in the access list are denied (dropped) or permitted (forwarded).

The *source-ip* parameter specifies the source IP address. Alternatively, you can specify the host name.

NOTE
To specify the host name instead of the IP address, the host name must be configured using the DNS resolver on the Brocade device. To configure the DNS resolver name, use the **ip dns server-address**... command at the global CONFIG level of the CLI.

The *wildcard* parameter specifies the mask value to compare against the host address specified by the *source-ip* parameter. The *wildcard* is in dotted-decimal notation (IP address format). It is a four-part value, where each part is 8 bits (one byte) separated by dots, and each bit is a one or a zero. Each part is a number ranging from 0 to 255, for example 0.0.0.255. Zeros in the mask mean the packet source address must match the *source-ip*. Ones mean any value matches. For example, the *source-ip* and *wildcard* values 10.157.22.26 0.0.0.255 mean that all hosts in the Class A subnet 10.157.22.x match the policy.

If you prefer to specify the wildcard (mask value) in CIDR format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of "10.157.22.26 0.0.0.255" as "10.157.22.26/24". The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into ones. For example, if you specify 10.157.22.26/24 or 10.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 10.157.22.0/24 (if you have enabled display of subnet lengths) or 10.157.22.0 0.0.0.255 in the startup-config file.

If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in "*/mask-bits*" format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

NOTE
If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show ip access-list** command.

The **host** *source-ip* | *hostname* parameter lets you specify a host IP address or name. When you use this parameter, you do not need to specify the mask. A mask of all zeros (0.0.0.0) is implied.

The **any** parameter configures the policy to match on all host addresses.

The **log** argument configures the device to generate syslog entries and SNMP traps for inbound packets that are denied by the access policy.

NOTE
You can enable logging on inbound ACLs and filters that support logging even when the ACLs and filters are already in use. To do so, re-enter the ACL or filter command and add the **log** parameter to the end of the ACL or filter. The software replaces the ACL or filter command with the new one. The new ACL or filter, with logging enabled, takes effect immediately.

The **in | out** parameter applies the ACL to incoming or outgoing traffic on the interface to which you apply the ACL. You can apply the ACL to an Ethernet port or virtual interface.

NOTE
If the ACL is bound to a virtual routing interface, you also can specify a subset of ports within the VLAN containing that interface when assigning an ACL to the interface. See "Enabling ACL filtering based on VLAN membership or VE port membership" on page 109 for further details.

Configuration example for standard named ACLs

To configure a standard named ACL, enter commands such as the following.

```
Brocade(config)# ip access-list standard Net1
Brocade(config-std-nACL)# deny host 10.157.22.26 log
Brocade(config-std-nACL)# deny 10.157.29.12 log
Brocade(config-std-nACL)# deny host IPHost1 log
Brocade(config-std-nACL)# permit any
Brocade(config-std-nACL)# exit
Brocade(config)# int ethernet 1/1/1
Brocade(config-if-e10000-1/1/1)# ip access-group Net1 in
```

The commands in this example configure a standard ACL named “Net1”. The entries in this ACL deny packets from three source IP addresses from being forwarded on port 1. Since the implicit action for an ACL is “deny”, the last ACL entry in this ACL permits all packets that are not explicitly denied by the first three ACL entries. For an example of how to configure the same entries in a numbered ACL, refer to “Configuring standard numbered ACLs” on page 86.

Notice that the command prompt changes after you enter the ACL type and name. The “std” in the command prompt indicates that you are configuring entries for a standard ACL. For an extended ACL, this part of the command prompt is “ext“. The “nACL” indicates that you are configuring a named ACL.

Extended numbered ACL configuration

This section describes how to configure extended numbered ACLs.

Extended ACLs let you permit or deny packets based on the following information:

- IP protocol
- Source IP address or host name
- Destination IP address or host name
- Source TCP or UDP port (if the IP protocol is TCP or UDP)
- Destination TCP or UDP port (if the IP protocol is TCP or UDP)

The IP protocol can be one of the following well-known names or any IP protocol number from 0 – 255:

- Internet Control Message Protocol (ICMP)
- Internet Group Management Protocol (IGMP)
- Internet Gateway Routing Protocol (IGRP)
- Internet Protocol (IP)
- Open Shortest Path First (OSPF)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

For TCP and UDP, you also can specify a comparison operator and port name or number. For example, you can configure a policy to block web access to a specific website by denying all TCP port 80 (HTTP) packets from a specified source IP address to the website IP address.

Extended numbered ACL syntax

Syntax: [**no**] **access-list** *ACL-num* **deny** | **permit** *ip-protocol source-ip* | *hostname wildcard* [*operator source-tcp/udp-port*] *destination-ip* | *hostname* [*icmp-num* | *icmp-type*] *wildcard* [*tcp/udp comparison operator destination-tcp/udp-port*] [**802.1p-priority-matching** <*0 –7*>] [**dscp-cos-mapping**] [**dscp-marking** <*0-63*> [**802.1p-priority-marking** <*0 –7*>**...** | **dscp-cos-mapping**]] [**dscp-matching** <*0-63*>] [**log**] [**precedence** *name* | <*0 – 7*>] [**tos** <*0 – 63*> | *name*] [**traffic policy** *name*]

Syntax: [**no**] **access-list** *ACL-num* **deny** | **permit host** *ip-protocol* **any any**

Syntax: [**no**] **ip access-group** *ACL-num* **in | out**

The *ACL-num* parameter is the extended access list number. Specify a number from 100–199.

The **deny** | **permit** parameter indicates whether packets that match the policy are dropped or forwarded.

The *ip-protocol* parameter indicates the type of IP packet you are filtering. You can specify a well-known name for any protocol whose number is less than 255. For other protocols, you must enter the number. Enter "?" instead of a protocol to list the well-known names recognized by the CLI.

The *source-ip* | *hostname* parameter specifies the source IP host for the policy. If you want the policy to match on all source addresses, enter **any**.

The *wildcard* parameter specifies the portion of the source IP host address to match against. The *wildcard* is in dotted-decimal notation (IP address format). It is a four-part value, where each part is 8 bits (one byte) separated by dots, and each bit is a one or a zero. Each part is a number ranging from 0 to 255, for example 0.0.0.255. Zeros in the mask mean the packet's source address must match the *source-ip* . Ones mean any value matches. For example, the *source-ip* and *wildcard* values 10.157.22.26 0.0.0.255 mean that all hosts in the Class A subnet 10.157.22.x match the policy.

If you prefer to specify the wildcard (mask value) in Classless Interdomain Routing (CIDR) format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of "10.157.22.26 0.0.0.255" as "10.157.22.26/24". The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into zeros. For example, if you specify 10.157.22.26/24 or 10.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 10.157.22.0/24 (if you have enabled display of subnet lengths) or 10.157.22.0 0.0.0.255 in the startup-config file.

If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in "*/ mask-bits* " format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

NOTE
If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show ip access-list** command.

The *destination-ip | hostname* parameter specifies the destination IP host for the policy. If you want the policy to match on all destination addresses, enter **any**.

The *icmp-type | icmp-num* parameter specifies the ICMP protocol type:

- This parameter applies only if you specified **icmp** as the *ip-protocol* value.
- If you use this parameter, the ACL entry is sent to the CPU for processing.
- If you do not specify a message type, the ACL applies to all types of ICMP messages.

The *icmp-num* parameter can be a value from 0–255.

The *icmp-type* parameter can have one of the following values, depending on the software version the device is running:

- any-icmp-type
- echo
- echo-reply
- information-request
- log
- mask-reply
- mask-request
- parameter-problem
- redirect
- source-quench
- time-exceeded
- timestamp-reply
- timestamp-request
- traffic policy
- unreachable
- *num*

NOTE
The QoS options listed below are only available if a specific ICMP type is specified for the *icmp-type* parameter and cannot be used with the **any-icmp-type** option above.

The *tcp/udp comparison operator* parameter specifies a comparison operator for the TCP or UDP port number. This parameter applies only when you specify **tcp** or **udp** as the IP protocol. For example, if you are configuring an entry for HTTP, specify **tcp eq http**. You can enter one of the following operators:

- **eq** – The policy applies to the TCP or UDP port name or number you enter after **eq**.
- **established** – This operator applies only to TCP packets. If you use this operator, the policy applies to TCP packets that have the ACK (Acknowledgment) or RST (Reset) bits set on (set to "1") in the Control Bits field of the TCP packet header. Thus, the policy applies only to established TCP sessions, not to new sessions. Refer to Section 3.1, "Header Format", in RFC 793 for information about this field.

NOTE
This operator applies only to destination TCP ports, not source TCP ports.

- **gt** – The policy applies to TCP or UDP port numbers greater than the port number or the numeric equivalent of the port name you enter after **gt**.
- **lt** – The policy applies to TCP or UDP port numbers that are less than the port number or the numeric equivalent of the port name you enter after **lt**.
- **neq** – The policy applies to all TCP or UDP port numbers except the port number or port name you enter after **neq**.
- **range** – The policy applies to all TCP or UDP port numbers that are between the first TCP or UDP port name or number and the second one you enter following the **range** parameter. The range includes the port names or numbers you enter. For example, to apply the policy to all ports between and including 23 (Telnet) and 53 (DNS), enter the following: **range 23 53**. The first port number in the range must be lower than the last number in the range.

The *tcp/udp-port* parameter specifies the TCP or UDP port number or well-known name. You can specify a well-known name for any application port whose number is less than 1024. For other application ports, you must enter the number. Enter "?" instead of a port to list the well-known names recognized by the CLI.

The **in** | **out** parameter specifies that the ACL applies to incoming traffic on the interface to which you apply the ACL. You can apply the ACL to an Ethernet port or a virtual interface.

NOTE
If the ACL is for a virtual routing interface, you also can specify a subset of ports within the VLAN containing that interface when assigning an ACL to the interface. Refer to "Configuring standard numbered ACLs" on page 86.

The **precedence** *name* | *num* parameter of the **ip access-list** command specifies the IP precedence. The precedence option for of an IP packet is set in a three-bit field following the four-bit header-length field of the packet's header. You can specify one of the following:

- **critical** or **5** – The ACL matches packets that have the critical precedence. If you specify the option number instead of the name, specify number 5.
- **flash** or **3** – The ACL matches packets that have the flash precedence. If you specify the option number instead of the name, specify number 3.
- **flash-override** or **4** – The ACL matches packets that have the flash override precedence. If you specify the option number instead of the name, specify number 4.
- **immediate** or **2** – The ACL matches packets that have the immediate precedence. If you specify the option number instead of the name, specify number 2.
- **internet** or **6** – The ACL matches packets that have the internetwork control precedence. If you specify the option number instead of the name, specify number 6.
- **network** or **7** – The ACL matches packets that have the network control precedence. If you specify the option number instead of the name, specify number 7.
- **priority** or **1** – The ACL matches packets that have the priority precedence. If you specify the option number instead of the name, specify number 1.
- **routine** or **0** – The ACL matches packets that have the routine precedence. If you specify the option number instead of the name, specify number 0.

The **tos** *name* | *num* parameter of the **ip access-list** command specifies the IP ToS. You can specify one of the following:

- **max-reliability** or **2** – The ACL matches packets that have the maximum reliability ToS. The decimal value for this option is 2.

- **max-throughput** or **4** – The ACL matches packets that have the maximum throughput ToS. The decimal value for this option is 4.
- **min-delay** or **8** – The ACL matches packets that have the minimum delay ToS. The decimal value for this option is 8.
- **min-monetary-cost** or **1** – The ACL matches packets that have the minimum monetary cost ToS. The decimal value for this option is 1.
  - **normal** or **0** – The ACL matches packets that have the normal ToS. The decimal value for this option is 0.
  - *num* – A number from 0 – 15 that is the sum of the numeric values of the options you want. The ToS field is a four-bit field following the Precedence field in the IP header. You can specify one or more of the following. To select more than one option, enter the decimal value that is equivalent to the sum of the numeric values of all the ToS options you want to select. For example, to select the **max-reliability** and **min-delay** options, enter number 10. To select all options, select 15.

NOTE
The following QoS options are only available if a specific ICMP type is specified and cannot be used with the **any-icmp-type** option set for the *icmp-type* parameter. See "QoS options for IP ACLs" on page 114 for more information on using ACLs to perform QoS.

The **802.1p-priority-matching** option inspects the 802.1p bit in the ACL that can be used with adaptive rate limiting. Enter a value from 0 – 7.

The **dscp-cos-mapping** option maps the DSCP value in incoming packets to a hardware table that provides mapping of each of the 0 – 63 DSCP values, and distributes them among eight traffic classes (internal priorities) and eight 802.1p priorities.

NOTE
The **dscp-cos-mapping** option overrides port-based priority settings.

NOTE
The **dscp-cos-mapping** option is not supported for Brocade ICX 6650 devices.

The **dscp-marking** option enables you to configure an ACL that marks matching packets with a specified DSCP value. Enter a value from 0 – 63. Refer to "Using an IP ACL to mark DSCP values (DSCP marking)" on page 115.

The **dscp-matching** option matches on the packet's DSCP value. Enter a value from 0 – 63. This option does not change the packet's forwarding priority through the device or mark the packet. Refer to "DSCP matching" on page 117.

The **log** parameter enables SNMP traps and Syslog messages for inbound packets denied by the ACL:

- You can enable logging on inbound ACLs and filters that support logging even when the ACLs and filters are already in use. To do so, re-enter the **ACL** or **filter** command and add the log parameter to the end of the ACL or filter. The software replaces the **ACL** or **filter** command with the new one. The new ACL or filter, with logging enabled, takes effect immediately.

The **traffic-policy** option enables the device to rate limit inbound traffic and to count the packets and bytes per packet to which ACL permit or deny clauses are applied. For configuration procedures and examples, refer to the chapter "ACL-based Rate Limiting" on page 141.

Configuration examples for extended numbered ACLs

To configure an extended access control list that blocks all Telnet traffic received on port 1/1/1 from IP host 10.157.22.26, enter the following commands.

```
Brocade(config)# access-list 101 deny tcp host 10.157.22.26 any eq telnet log
Brocade(config)# access-list 101 permit ip any any
Brocade(config)# interface ethernet 1/1/1
Brocade(config-if-e10000-1/1/1)# ip access-group 101 in
Brocade(config)# write memory
```

Here is another example of commands for configuring an extended ACL and applying it to an interface. These examples show many of the syntax choices. Notice that some of the entries are configured to generate log entries while other entries are not thus configured.

```
Brocade(config)# access-list 102 perm icmp 10.157.22.0/24 209.157.21.0/24
Brocade(config)# access-list 102 deny igmp host rkwong 10.157.21.0/24 log
Brocade(config)# access-list 102 deny igrp 10.157.21.0/24 host rkwong log
Brocade(config)# access-list 102 deny ip host 10.157.21.100 host 10.157.22.1 log
Brocade(config)# access-list 102 deny ospf any any log
Brocade(config)# access-list 102 permit ip any any
```

The first entry permits ICMP traffic from hosts in the 10.157.22.x network to hosts in the 10.157.21.x network.

The second entry denies IGMP traffic from the host device named “rkwong” to the 10.157.21.x network.

The third entry denies IGMP traffic from the 10.157.21.x network to the host device named “rkwong”.

The fourth entry denies all IP traffic from host 10.157.21.100to host 10.157.22.1 and generates syslog entries for packets that are denied by this entry.

The fifth entry denies all OSPF traffic and generates syslog entries for denied traffic.

The sixth entry permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL would deny all incoming or outgoing IP traffic on the ports to which you assign the ACL.

The following commands apply ACL 102 to the incoming traffic on port 1/1/2 and to the incoming traffic on port 1/3/1.

```
Brocade(config)# interface ethernet 1/1/2
Brocade(config-if-e10000-1/1/2)# ip access-group 102 in
Brocade(config-if-e10000-1/1/2)# exit
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# ip access-group 102 in
Brocade(config)# write memory
```

Here is another example of an extended ACL.

```
Brocade(config)# access-list 103 deny tcp 10.157.21.0/24 10.157.22.0/24
Brocade(config)# access-list 103 deny tcp 10.157.21.0/24 eq ftp 10.157.22.0/24
Brocade(config)# access-list 103 deny tcp 10.157.21.0/24 10.157.22.0/24 lt
telnet neq 5
Brocade(config)# access-list 103 deny udp any range 5 6 10.157.22.0/24 range 7 8
Brocade(config)# access-list 103 permit ip any any
```

The first entry in this ACL denies TCP traffic from the 10.157.21.x network to the 10.157.22.x network.

The second entry denies all FTP traffic from the 10.157.21.x network to the 10.157.22.x network.

The third entry denies TCP traffic from the 10.157.21.x network to the 10.157.22.x network, if the TCP port number of the traffic is less than the well-known TCP port number for Telnet (23), and if the TCP port is not equal to 5. Thus, TCP packets whose TCP port numbers are 5 or are greater than 23 are allowed.

The fourth entry denies UDP packets from any source to the 10.157.22.x network, if the UDP port number from the source network is 5 or 6 and the destination UDP port is 7 or 8.

The fifth entry permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL would deny all incoming or outgoing IP traffic on the ports to which you assign the ACL.

The following commands apply ACL 103 to the incoming traffic on ports 1/2/1 and 1/2/2.

```
Brocade(config)# interface ethernet 1/2/1
Brocade(config-if-e10000-1/2/1)# ip access-group 103 in
Brocade(config-if-e10000-1/2/1)# exit
Brocade(config)# interface ethernet 1/2/2
Brocade(config-if-e10000-1/2/2)# ip access-group 103 in
Brocade(config)# write memory
```

Extended named ACL configuration

The commands for configuring named ACL entries are different from the commands for configuring numbered ACL entries. The command to configure a numbered ACL is **access-list**. The command for configuring a named ACL is **ip access-list**. In addition, when you configure a numbered ACL entry, you specify all the command parameters on the same command. When you configure a named ACL, you specify the ACL type (standard or extended) and the ACL number with one command, which places you in the configuration level for that ACL. Once you enter the configuration level for the ACL, the command syntax is the same as the syntax for numbered ACLs.

Extended ACLs let you permit or deny packets based on the following information:

- IP protocol
- Source IP address or host name
- Destination IP address or host name
- Source TCP or UDP port (if the IP protocol is TCP or UDP)
- Destination TCP or UDP port (if the IP protocol is TCP or UDP)

The IP protocol can be one of the following well-known names or any IP protocol number from 0 – 255:

- Internet Control Message Protocol (ICMP)
- Internet Group Management Protocol (IGMP)
- Internet Gateway Routing Protocol (IGRP)
- Internet Protocol (IP)
- Open Shortest Path First (OSPF)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

For TCP and UDP, you also can specify a comparison operator and port name or number. For example, you can configure a policy to block web access to a specific website by denying all TCP port 80 (HTTP) packets from a specified source IP address to the website's IP address.

Extended named ACL syntax

Syntax: [**no**] **ip access-list extended** *ACL-name* **deny** | **permit** *ip-protocol source-ip* | *hostname wildcard* [*operator source-tcp/udp-port*] *destination-ip* | *hostname* [*icmp-num* | *icmp-type*] *wildcard* [*tcp/udp comparison operator destination-tcp/udp-port*] [**802.1p-priority-matching** <*0 –7*>] [**dscp-cos-mapping**] [**dscp-marking** <*0-63*> [**802.1p-priority-marking** <*0 –7*>**...** | **dscp-cos-mapping**]] [**dscp-matching** <*0-63*>] [**log**] [**precedence** *name* | <*0 – 7*>] [**tos** <*0 – 63*> | *name*] [**traffic policy** *name*]

The *ACL-name* parameter is the access list name. You can specify a string of up to 256 alphanumeric characters. You can use blanks in the ACL name if you enclose the name in quotation marks (for example, "ACL for Net1").

The **deny** | **permit** parameter indicates whether packets that match the policy are dropped or forwarded.

The *ip-protocol* parameter indicates the type of IP packet you are filtering. You can specify a well-known name for any protocol whose number is less than 255. For other protocols, you must enter the number. Enter "?" instead of a protocol to list the well-known names recognized by the CLI.

The *source-ip* | *hostname* parameter specifies the source IP host for the policy. If you want the policy to match on all source addresses, enter **any**.

The *wildcard* parameter specifies the portion of the source IP host address to match against. The *wildcard* is in dotted-decimal notation (IP address format). It is a four-part value, where each part is 8 bits (one byte) separated by dots, and each bit is a one or a zero. Each part is a number ranging from 0 to 255, for example 0.0.0.255. Zeros in the mask mean the packet's source address must match the *source-ip* . Ones mean any value matches. For example, the *source-ip* and *wildcard* values 10.157.22.26 0.0.0.255 mean that all hosts in the Class A subnet 10.157.22.x match the policy.

If you prefer to specify the wildcard (mask value) in Classless Interdomain Routing (CIDR) format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of "10.157.22.26 0.0.0.255" as "10.157.22.26/24". The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into zeros. For example, if you specify 10.157.22.26/24 or 10.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 10.157.22.0/24 (if you have enabled display of subnet lengths) or 10.157.22.0 0.0.0.255 in the startup-config file.

If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in "/ *mask-bits* " format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

NOTE
If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show ip access-list** command.

The *destination-ip* | *hostname* parameter specifies the destination IP host for the policy. If you want the policy to match on all destination addresses, enter **any**.

The *icmp-type* | *icmp-num* parameter specifies the ICMP protocol type:

- This parameter applies only if you specified **icmp** as the *ip-protocol* value.
- If you use this parameter, the ACL entry is sent to the CPU for processing.
- If you do not specify a message type, the ACL applies to all types of ICMP messages.

The *icmp-num* parameter can be a value from 0 – 255.

The *icmp-type* parameter can have one of the following values, depending on the software version the device is running:

- any-icmp-type
- echo
- echo-reply
- information-request
- log
- mask-reply
- mask-request
- parameter-problem
- redirect
- source-quench
- time-exceeded
- timestamp-reply
- timestamp-request
- traffic policy
- unreachable
- *num*

NOTE
The QoS options listed below are only available if a specific ICMP type is specified for the *icmp-type* parameter and cannot be used with the **any-icmp-type** option above. See "QoS options for IP ACLs" on page 1734for more information on using ACLs to perform QoS.

The *tcp/udp comparison operator* parameter specifies a comparison operator for the TCP or UDP port number. This parameter applies only when you specify **tcp** or **udp** as the IP protocol. For example, if you are configuring an entry for HTTP, specify **tcp eq http**. You can enter one of the following operators:

- **eq** – The policy applies to the TCP or UDP port name or number you enter after **eq**.
- **established** – This operator applies only to TCP packets. If you use this operator, the policy applies to TCP packets that have the ACK (Acknowledgment) or RST (Reset) bits set on (set to "1") in the Control Bits field of the TCP packet header. Thus, the policy applies only to established TCP sessions, not to new sessions. Refer to Section 3.1, "Header Format", in RFC 793 for information about this field.

NOTE
This operator applies only to destination TCP ports, not source TCP ports.

- **gt** – The policy applies to TCP or UDP port numbers greater than the port number or the numeric equivalent of the port name you enter after **gt**.
- **lt** – The policy applies to TCP or UDP port numbers that are less than the port number or the numeric equivalent of the port name you enter after **lt**.
- **neq** – The policy applies to all TCP or UDP port numbers except the port number or port name you enter after **neq**.
- **range** – The policy applies to all TCP or UDP port numbers that are between the first TCP or UDP port name or number and the second one you enter following the **range** parameter. The range includes the port names or numbers you enter. For example, to apply the policy to all ports between and including 23 (Telnet) and 53 (DNS), enter the following: **range 23 53**. The first port number in the range must be lower than the last number in the range.

The *tcp/udp-port* parameter specifies the TCP or UDP port number or well-known name. You can specify a well-known name for any application port whose number is less than 1024. For other application ports, you must enter the number. Enter "?" instead of a port to list the well-known names recognized by the CLI.

The **in** | **out** parameter specifies that the ACL applies to incoming traffic on the interface to which you apply the ACL. You can apply the ACL to an Ethernet port or a virtual interface.

NOTE
If the ACL is for a virtual routing interface, you also can specify a subset of ports within the VLAN containing that interface when assigning an ACL to the interface. Refer to "Configuring standard numbered ACLs" on page 1703.

The **precedence** *name* | *num* parameter of the **ip access-list** command specifies the IP precedence. The precedence option for of an IP packet is set in a three-bit field following the four-bit header-length field of the packet's header. You can specify one of the following:

- **critical** or **5** – The ACL matches packets that have the critical precedence. If you specify the option number instead of the name, specify number 5.
- **flash** or **3** – The ACL matches packets that have the flash precedence. If you specify the option number instead of the name, specify number 3.
- **flash-override** or **4** – The ACL matches packets that have the flash override precedence. If you specify the option number instead of the name, specify number 4.
- **immediate** or **2** – The ACL matches packets that have the immediate precedence. If you specify the option number instead of the name, specify number 2.

- **internet** or **6** – The ACL matches packets that have the internetwork control precedence. If you specify the option number instead of the name, specify number 6.
- **network** or **7** – The ACL matches packets that have the network control precedence. If you specify the option number instead of the name, specify number 7.
- **priority** or **1** – The ACL matches packets that have the priority precedence. If you specify the option number instead of the name, specify number 1.
- **routine** or **0** – The ACL matches packets that have the routine precedence. If you specify the option number instead of the name, specify number 0.

The **tos** *name* | *num* parameter of the **ip access-list** command specifies the IP ToS. You can specify one of the following:

- **max-reliability** or **2** – The ACL matches packets that have the maximum reliability ToS. The decimal value for this option is 2.
- **max-throughput** or **4** – The ACL matches packets that have the maximum throughput ToS. The decimal value for this option is 4.
- **min-delay** or **8** – The ACL matches packets that have the minimum delay ToS. The decimal value for this option is 8.
- **min-monetary-cost** or **1** – The ACL matches packets that have the minimum monetary cost ToS. The decimal value for this option is 1.

  NOTE
  This value is not supported on 10 Gigabit Ethernet modules.

  - **normal** or **0** – The ACL matches packets that have the normal ToS. The decimal value for this option is 0.
  - *num* – A number from 0 – 15 that is the sum of the numeric values of the options you want. The ToS field is a four-bit field following the Precedence field in the IP header. You can specify one or more of the following. To select more than one option, enter the decimal value that is equivalent to the sum of the numeric values of all the ToS options you want to select. For example, to select the **max-reliability** and **min-delay** options, enter number 10. To select all options, select 15.

NOTE
The following QoS options are only available if a specific ICMP type is specified and cannot be used with the **any-icmp-type** option set for the *icmp-type* parameter. See "QoS options for IP ACLs" on page 1734 for more information on using ACLs to perform QoS.

The **802.1p-priority-matching** option inspects the 802.1p bit in the ACL that can be used with adaptive rate limiting. Enter a value from 0 – 7. For details, refer to "Inspecting the 802.1p bit in the ACL for adaptive rate limiting" on page 1773.

The **dscp-cos-mapping** option maps the DSCP value in incoming packets to a hardware table that provides mapping of each of the 0 – 63 DSCP values, and distributes them among eight traffic classes (internal priorities) and eight 802.1p priorities.

NOTE
The **dscp-cos-mapping** option overrides port-based priority settings.

The **dscp-marking** option enables you to configure an ACL that marks matching packets with a specified DSCP value. Enter a value from 0 – 63. Refer to "Using an IP ACL to mark DSCP values (DSCP marking)" on page 1736.

The **dscp-matching** option matches on the packet's DSCP value. Enter a value from 0 – 63. This option does not change the packet's forwarding priority through the device or mark the packet. Refer to "DSCP matching" on page 1738.

The **log** parameter enables SNMP traps and Syslog messages for inbound packets denied by the ACL:

- You can enable logging on inbound ACLs and filters that support logging even when the ACLs and filters are already in use. To do so, re-enter the **ACL** or **filter** command and add the log parameter to the end of the ACL or filter. The software replaces the **ACL** or **filter** command with the new one. The new ACL or filter, with logging enabled, takes effect immediately.

The **traffic-policy** option enables the device to rate limit inbound traffic and to count the packets and bytes per packet to which ACL permit or deny clauses are applied. For configuration procedures and examples, refer to the chapter "Traffic Policies" on page 1765.

Configuration example for extended named ACLs

To configure an extended named ACL, enter the **ip access-list extended** *ACL_name* command.

```
Brocade(config)# ip access-list extended "block Telnet"
Brocade(config-ext-nACL)# deny tcp host 10.157.22.26 any eq telnet log
Brocade(config-ext-nACL)# permit ip any any
Brocade(config-ext-nACL)# exit
Brocade(config)# interface ethernet 1/1/1
Brocade(config-if-e10000-1/1/1)# ip access-group "block Telnet" in
```

The options at the ACL configuration level and the syntax for the **ip access-group** command are the same for numbered and named ACLs and are described in "Extended numbered ACL configuration" on page 90 and "Extended numbered ACL configuration" on page 90.

Applying egress ACLs to Control (CPU) traffic

By default, outbound ACLs are not applied to traffic generated by the CPU. This must be enabled using the **enable egress-acl-on-cpu-traffic** command.

Syntax: enable egress-acl-on-cpu-traffic

Preserving user input for ACL TCP/UDP port numbers

ACL implementations automatically display the TCP/UDP port name instead of the port number, regardless of user preference, unless the device is configured to preserve user input. When the option to preserve user input is enabled, the system will display either the port name or the number.

To enable this feature, enter the **ip preserve-ACL-user-input-format** command.

```
Brocade(config)# ip preserve-ACL-user-input-format
```

Syntax: ip preserve-ACL-user-input-format

The following example shows how this feature works for a TCP port (this feature works the same way for UDP ports). In this example, the user identifies the TCP port by number (80) when configuring ACL group 140. However, **show ip access-list 140** reverts back to the port name for the TCP port (http in this example). After the user issues the new **ip preserve-ACL-user-input-format** command, **show ip access-list 140** displays either the TCP port number or name, depending on how it was configured by the user.

```
Brocade(config)# access-list 140 permit tcp any any eq 80
Brocade(config)# access-list 140 permit tcp any any eq ftp
Brocade# show ip access-lists 140
Extended IP access list 140
permit tcp any any eq http
permit tcp any any eq ftp
Brocade(config)#ip preserve-ACL-user-input-format
Brocade#show ip access-lists 140
Extended IP access list 140
permit tcp any any eq 80
permit tcp any any eq ftp
```

ACL comment text management

ACL comment text describes entries in an ACL. The comment text appears in the output of **show** commands that display ACL information.

This section describes how to add, delete, and view ACL comments.

Adding a comment to an entry in a numbered ACL

To add comments to entries in a numbered ACL, enter commands such as the following.

```
Brocade(config)# access-list 100 remark The following line permits TCP packets
Brocade(config)# access-list 100 permit tcp 192.168.4.40/24 10.2.2.2/24
Brocade(config)# access-list 100 remark The following permits UDP packets
Brocade(config)# access-list 100 permit udp 192.168.2.52/24 10.2.2.2/24
Brocade(config)# access-list 100 deny ip any any
```

You can add comments to entries in a numbered ACL using the syntax for named ACLs. For example, using the same example configuration above, you could instead enter the following commands.

```
Brocade(config)# ip access-list extended 100
Brocade(config-ext-nACL)# remark The following line permits TCP packets
Brocade(config-ext-nACL)# permit tcp 192.168.4.40/24 10.2.2.2/24
Brocade(config-ext-nACL)# remark The following permits UDP packets
Brocade(config-ext-nACL)# permit udp 192.168.2.52/24 10.2.2.2/24
Brocade(config-ext-nACL)# deny ip any any
```

Syntax: [**no**] **access-list** *ACL-num* **remark** *comment-text*

or

Syntax: [**no**] **ip access-list standard** | **extended** *ACL-num*

Syntax: **remark** *comment-text*

For *ACL-num*, enter the number of the ACL.

The *comment-text* can be up to 128 characters in length. The comment must be entered separately from the actual ACL entry; that is, you cannot enter the ACL entry and the ACL comment with the same **access-list** or **ip access-list** command. Also, in order for the remark to be displayed correctly in the output of **show** commands, the comment must be entered immediately before the ACL entry it describes. Note that an ACL comment is tied to the ACL entry immediately following the comment. Therefore, if the ACL entry is removed, the ACL comment is also removed.

The **standard** | **extended** parameter indicates the ACL type.

Adding a comment to an entry in a named ACL

To add comments to entries in a named ACL, enter commands such as the following.

```
Brocade(config)# ip access-list extended TCP/UDP
Brocade(config-ext-nACL)# remark The following line permits TCP packets
Brocade(config-ext-nACL)# permit tcp 192.168.4.40/24 10.2.2.2/24
Brocade(config-ext-nACL)# remark The following permits UDP packets
Brocade(config-ext-nACL)# permit udp 192.168.2.52/24 10.2.2.2/24
Brocade(config-ext-nACL)# deny ip any any
```

Syntax: [**no**] **access-list standard** | **extended** *ACL-name*

Syntax: **remark** *comment-text*

The **standard** | **extended** parameter indicates the ACL type.

For *ACL-name*, enter the name of the ACL.

The *comment-text* can be up to 128 characters in length. The comment must be entered separately from the actual ACL entry; that is, you cannot enter the ACL entry and the ACL comment with the same **ip access-list** command. Also, in order for the remark to be displayed correctly in the output of show commands, the comment must be entered immediately before the ACL entry it describes. Note that an ACL comment is tied to the ACL entry immediately following the comment. Therefore, if the ACL entry is removed, the ACL comment is also removed.

Deleting a comment from an ACL entry

To delete a comment from an ACL entry, enter commands such as the following.

```
Brocade(config)# ip access-list standard 99
Brocade(config)# no remark The following line permits TCP packets
```

Syntax: **no remark** *comment-text*

Viewing comments in an ACL

You can use the following commands to display comments for ACL entries:

- **show running-config**
- **show access-list**
- **show ip access-list**

The following shows the comment text for a numbered ACL, ACL 100, in a **show running-config** display.

```
Brocade# show running-config
…
access-list 100 remark The following line permits TCP packets
access-list 100 permit tcp 192.168.4.40/24 10.2.2.2/24
access-list 100 remark The following line permits UDP packets
access-list 100 permit udp 192.168.2.52/24 10.2.2.2/24
access-list 100 deny ip any any
```

Syntax: show running-config

The following example shows the comment text for an ACL in a **show access-list** display. The output is identical in a **show ip access-list** display.

```
Brocade# show access-list
IP access list rate-limit 100 0000.00bb.cccc
Extended IP access list TCP/UDP (Total flows: N/A, Total packets: N/A)
ACL Remark:  The following line permits TCP packets
permit tcp 10.0.0.40 255.255.255.0 10.0.0.2 255.255.255.0 (Flows: N/A, Packets:
N/A)
ACL Remark:  The following line permits UDP packets
permit udp 10.0.0.52 255.255.255.0 10.0.0.2 255.255.255.0 (Flows: N/A, Packets:
N/A)
deny ip any any (Flows: N/A, Packets: N/A)
```

Syntax: show access-list *ACL-num* | *ACL-name* | **all**

or

Syntax: show ip access-list *ACL-num* | *ACL-name* | **all**

Applying an ACL to a virtual interface in a protocol- or subnet-based VLAN

By default, when you apply an ACL to a virtual interface in a protocol-based or subnet-based VLAN, the ACL takes effect on all protocol or subnet VLANs to which the untagged port belongs. To prevent the Brocade device from denying packets on other virtual interfaces that do not have an ACL applied, configure an ACL that permits packets in the IP subnet of the virtual interface in all protocol-based or subnet-based VLANs to which the untagged port belongs. The following is an example configuration.

```
Brocade# configure terminal
Brocade(config)# vlan 1 name DEFAULT-VLAN by port
Brocade(config-vlan-1)# ip-subnet 192.168.10.0 255.255.255.0
Brocade(config-vlan-ip-subnet)# static ethe 1
Brocade(config-vlan-ip-subnet)# router-interface ve 10
Brocade(config-vlan-ip-subnet)# ip-subnet 10.15.1.0 255.255.255.0
Brocade(config-vlan-ip-subnet)# static ethe 1/1/1
Brocade(config-vlan-ip-subnet)# router-interface ve 20
Brocade(config-vlan-ip-subnet)# logging console
Brocade(config-vlan-ip-subnet)# exit
```

```
Brocade(config-vlan-1)# no vlan-dynamic-discovery
  Vlan dynamic discovery is disabled
Brocade(config-vlan-1)# interface ethernet 1/1/2
Brocade(config-if-e1000-2)# disable
Brocade(config-if-e1000-2)# interface ve 10
Brocade(config-vif-10)# ip address 192.168.10.254 255.255.255.0
Brocade(config-vif-10)# interface ve 20
Brocade(config-vif-20)# ip access-group test1 in
Brocade(config-vif-20)# ip address 10.15.1.10 255.255.255.0
Brocade(config-vif-20)# exit
Brocade(config)# ip access-list extended test1
Brocade(config-ext-nACL)# permit ip 10.15.1.0 0.0.0.255 any log
Brocade(config-ext-nACL)# permit ip 192.168.10.0 0.0.0.255 any log
Brocade(config-ext-nACL)# end
Brocade#
```

ACL logging

Brocade devices support ACL logging of inbound packets that are sent to the CPU for processing (denied packets).

NOTE
ACL logging is not supported for outbound packets or any packets that are processed in hardware (permitted packets).

You may want the software to log entries in the syslog for packets that are denied by ACL filters. ACL logging is disabled by default; it must be explicitly enabled on a port.

When you enable logging for ACL entries, statistics for packets that match the deny conditions of the ACL entries are logged. For example, if you configure a standard ACL entry to deny all packets from source address 10.157.22.26, statistics for packets that are explicitly denied by the ACL entry are logged in the Syslog buffer and in SNMP traps sent by the Brocade device.

The first time an ACL entry denies a packet, the software immediately generates a Syslog entry and an SNMP trap. The software also starts a five-minute timer. The timer keeps track of all packets explicitly denied by the ACL entries. After five minutes, the software generates a single Syslog entry for each ACL entry that denied a packet. The Syslog entry (message) indicates the number of packets denied by the ACL entry during the previous five minutes. Note however that packet count may be inaccurate if the packet rate is high and exceeds the CPU processing rate.

If no ACL entries explicitly deny packets during an entire five-minute timer interval, the timer stops. The timer restarts when an ACL entry explicitly denies a packet.

NOTE
The timer for logging packets denied by MAC address filters is a different timer than the ACL logging timer.

Configuration notes for ACL logging

Note the following points before configuring ACL logging:

- ACL logging is supported for denied packets, which are sent to the CPU for logging. ACL logging is not supported for permitted packets.

- ACL logging is not supported for dynamic ACLs with multi-device port authentication and 802.1X.
- Packets that are denied by ACL filters are logged in the Syslog based on a sample time-period.
- You can enable ACL logging on physical and virtual interfaces.
- When ACL logging is disabled, packets that match the ACL rule are forwarded or dropped in hardware.
- ACL logging is supported for ACLs that are applied to network management access features such as Telnet, SSH, and SNMP.
- When an ACL that includes an entry with a logging option is applied to a port that has logging enabled, if an ACL that includes an entry with a logging option is applied to another port in the same port region, then traffic on the latter port is also logged, whether logging is explicitly enabled for that latter port or not. If logging is enabled on multiple ports in the same port region, then logging will only be disabled if it is disabled on all the ports in the same port region.

NOTE
The above limitation applies only to IPv4 ACLs, it does not apply to the use of ACLs to log IPv6 traffic.

- When ACL logging is enabled, packets sent to the CPU are automatically rate limited to prevent CPU overload.
- ACL logging is intended for debugging purposes. Brocade recommends that you disable ACL logging after the debug session is over.

Configuration tasks for ACL logging

To enable ACL logging, complete the following steps:

1. Create ACL entries with the log option
2. Enable ACL logging on individual ports

NOTE
The command syntax for enabling ACL logging is different on IPv4 devices than on IPv6 devices. See the configuration examples in the next section.

3. Bind the ACLs to the ports on which ACL logging is enabled

Example ACL logging configuration

The following shows an example ACL logging configuration on an IPv4 device.

```
Brocade(config)# access-list 1 deny host 10.157.22.26 log
Brocade(config)# access-list 1 deny 10.157.29.12 log
Brocade(config)# access-list 1 deny host IPHost1 log
Brocade(config)# access-list 1 permit any
Brocade(config)# interface ethernet 1/1/4
Brocade(config-if-e10000-1/1/4)# ACL-logging
Brocade(config-if-e10000-1/1/4)# ip access-group 1 in
```

The above commands create ACL entries that include the log option, enable ACL logging on interface e 1/1/4, then bind the ACL to interface e 1/1/4. Statistics for packets that match the deny statements will be logged.

Syntax: ACL-logging

The **ACL-logging** command applies to IPv4 devices only. For IPv6 devices, use the **logging-enable** command as shown in the following example.

The following shows an example configuration on an IPv6 device.

```
Brocade(config)# ipv6 access-list ACL_log_v6
Brocade(config-ipv6-access-list ACL_log_v6)# logging-enable
Brocade(config-ipv6-access-list ACL_log_v6)# deny ipv6 host 2001:DB8::1 any log
Brocade(config-ipv6-access-list ACL_log_v6)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# ipv6 traffic-filter ACL_log_v6 in
```

The above commands create ACL entries that include the log option, then bind the ACL to interface e 1/3/1. Statistics for packets that match the deny statement will be logged.

Syntax: logging-enable

NOTE
The **logging-enabled** command applies to IPv6 devices only. For IPv4 devices, use the **ACL-logging** command as shown in the previous example.

Displaying ACL Log Entries

The first time an entry in an ACL permits or denies a packet and logging is enabled for that entry, the software generates a Syslog message and an SNMP trap. Messages for packets permitted or denied by ACLs are at the warning level of the Syslog.

When the first Syslog entry for a packet permitted or denied by an ACL is generated, the software starts an ACL timer. After this, the software sends Syslog messages every five minutes. If an ACL entry does not permit or deny any packets during the timer interval, the software does not generate a Syslog entry for that ACL entry.

NOTE
For an ACL entry to be eligible to generate a Syslog entry for denied packets, logging must be enabled for the entry. The Syslog contains entries only for the ACL entries that deny packets and have logging enabled.

To display syslog entries, enter the **show log** command from any CLI prompt:

```
Brocade# show log
Syslog logging: enabled (0 messages dropped, 2 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 9 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Dynamic Log Buffer (50 lines):
0d00h12m18s:W:ACL: ACL: List 122 denied tcp 10.20.15.6(0)(Ethernet 4 0000.0004.01
10.20.18.6(0), 1 event(s)
0d00h12m18s:W:ACL: ACL: List 122 denied tcp 10.20.15.2(0)(Ethernet 4 0000.0004.01
10.20.18.2(0), 1 event(s)
0d00h12m18s:W:ACL: ACL: List 122 denied tcp 10.20.15.4(0)(Ethernet 4 0000.0004.01
10.20.18.4(0), 1 event(s)
0d00h12m18s:W:ACL: ACL: List 122 denied tcp 10.20.15.3(0)(Ethernet 4 0000.0004.01
10.20.18.3(0), 1 event(s)
0d00h12m18s:W:ACL: ACL: List 122 denied tcp 10.20.15.5(0)(Ethernet 4 0000.0004.01
10.20.18.5(0), 1 event(s)
0d00h12m18s:I:ACL: 122 applied to port 4 by  from console session
0d00h10m12s:I:ACL: 122 removed from port 4 by  from console session
0d00h09m56s:I:ACL: 122 removed from port 4 by  from console session
0d00h09m38s:I:ACL: 122 removed from port 4 by  from console session
```

Syntax: show log

Enabling strict control of ACL filtering of fragmented packets

The default processing of fragments by hardware-based ACLs is as follows:

- The first fragment of a packet is permitted or denied using the ACLs. The first fragment is handled the same way as non-fragmented packets, since the first fragment contains the Layer 4 source and destination application port numbers. The device uses the Layer 4 CAM entry if one is programmed, or applies the interface's ACL entries to the packet and permits or denies the packet according to the first matching ACL.
- For other fragments of the same packet, they are subject to a rule only if there is no Layer 4 information in the rule or in any preceding rules.

The fragments are forwarded even if the first fragment, which contains the Layer 4 information, was denied. Generally, denying the first fragment of a packet is sufficient, since a transaction cannot be completed without the entire packet.

For tighter control, you can configure the port to drop all packet fragments. To do so, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/1/1
Brocade(config-if-e10000-1/1/1)# ip access-group frag deny
```

This option begins dropping all fragments received by the port as soon as you enter the command. This option is especially useful if the port is receiving an unusually high rate of fragments, which can indicate a hacker attack.

Syntax: [no] ip access-group frag deny

Enabling ACL support for switched traffic in the router image

For Brocade ICX 6650, ACL support for switched traffic in the router image is enabled by default. There is no command to enable or disable it.

For outbound traffic, ACL support is enabled on switched traffic by default. The **bridged-routed** command is not applicable.

Enabling ACL filtering based on VLAN membership or VE port membership

NOTE
This section applies to IPv4 ACLs only. IPv6 ACLs do not support ACL filtering based on VLAN membership or VE port membership.

This feature is not applicable to outbound traffic.

You can apply an inbound IPv4 ACL to specific VLAN members on a port (Layer 2 devices only) or to specific ports on a virtual interface (VE) (Layer 3 Devices only). By default, this feature support is disabled. To enable it, enter the following commands at the Global CONFIG level of the CLI.

```
Brocade(config)# enable ACL-per-port-per-vlan
Brocade(config)# write memory
Brocade(config)# exit
Brocade# reload
```

NOTE
For complete configuration examples, see "Applying an IPv4 ACL to specific VLAN members on a port (Layer 2 devices only)" on page 110 and "Applying an IPv4 ACL to a subset of ports on a virtual interface (Layer 3 devices only)" on page 110.

Syntax: [no] **enable ACL-per-port-per-vlan**

Enter the **no** form of the command to disable this feature.

Configuration notes for ACL filtering

- Before enabling this feature on an IPv4 device, make sure the VLAN numbers are contiguous. For example, the VLAN numbers can be 201, 202, 203, and 204, but not 300, 401, 600, and 900.
- Brocade devices do not support a globally-configured PBR policy together with per-port-per-VLAN ACLs.
- IPv4 ACLs that filter based on VLAN membership or VE port membership (ACL-per-port-per-VLAN), are supported together with IPv6 ACLs on the same device, as long as they are not bound to the same port or virtual interface.

Applying an IPv4 ACL to specific VLAN members on a port (Layer 2 devices only)

NOTE
This section applies to IPv4 ACLs only. IPv6 ACLs do not support ACL filtering based on VLAN membership.

When you bind an IPv4 ACL to a port, the port filters all inbound traffic on the port. However, on a tagged port, there may be a need to treat packets for one VLAN differently from packets for another VLAN. In this case, you can configure a tagged port on a Layer 2 device to filter packets based on the packets' VLAN membership.

To apply an IPv4 ACL to a specific VLAN on a port, enter commands such as the following.

```
Brocade(config)# enable ACL-per-port-per-vlan
...
Brocade(config)# vlan 12 name vlan12
Brocade(config-vlan-12)# untag ethernet 1/1/5 to 1/1/8
Brocade(config-vlan-12)# tag ethernet 1/1/23 to 1/1/24
Brocade(config-vlan-12)# exit
Brocade(config)# access-list 10 deny host 10.157.22.26 log
Brocade(config)# access-list 10 deny 10.157.29.12 log
Brocade(config)# access-list 10 deny host IPHost1 log
Brocade(config)# access-list 10 permit
Brocade(config)# interface ethernet 1/1/23
Brocade(config-if-e10000-1/1/23))# per-vlan 12
Brocade(config-if-e10000-1/1/23-vlan-12))# ip access-group 10 in
```

The commands in this example configure port-based VLAN 12, and add ports e1/1/ 5 –1/1/ 8 as untagged ports and ports e 1/1/23 – 1/1/24 as tagged ports to the VLAN. The commands following the VLAN configuration commands configure ACL 10. Finally, the last three commands apply ACL 10 on VLAN 12 for which port e 1/1/23 is a member.

Syntax: **per-vlan** *VLAN ID*

Syntax: [**no**] **ip access-group** *ACL ID*

The *VLAN ID* parameter specifies the VLAN name or number to which you will bind the ACL.

The *ACL ID* parameter is the access list name or number.

Applying an IPv4 ACL to a subset of ports on a virtual interface (Layer 3 devices only)

NOTE
This section applies to IPv4 ACLs only. IPv6 ACLs do not support ACL filtering based on VE port membership.

You can apply an IPv4 ACL to a virtual routing interface. The virtual interface is used for routing between VLANs and contains all the ports within the VLAN. The IPv4 ACL applies to all the ports on the virtual routing interface. You also can specify a subset of ports within the VLAN containing a specified virtual interface when assigning an ACL to that virtual interface.

Use this feature when you do not want the IPv4 ACLs to apply to all the ports in the virtual interface VLAN or when you want to streamline IPv4 ACL performance for the VLAN.

To apply an ACL to a subset of ports within a virtual interface, enter commands such as the following.

```
Brocade(config)# enable ACL-per-port-per-vlan
Brocade(config)# vlan 10 name IP-subnet-vlan
Brocade(config-vlan-10)# untag ethernet 1/1/1 to 1/2/12
Brocade(config-vlan-10)# router-interface ve 1
Brocade(config-vlan-10)# exit
Brocade(config)# access-list 1 deny host 10.157.22.26 log
Brocade(config)# access-list 1 deny 10.157.29.12 log
Brocade(config)# access-list 1 deny host IPHost1 log
Brocade(config)# access-list 1 permit any
Brocade(config)# interface ve 1
Brocade(config-vif-1/1)# ip access-group 1 in ethernet 1/1/1 ethernet 1/1/3
ethernet 1/2/1 to 1/2/4
```

The commands in this example configure port-based VLAN 10, add ports 1/1/1 –1/2/12 to the VLAN, and add virtual routing interface 1 to the VLAN. The commands following the VLAN configuration commands configure ACL 1. Finally, the last two commands apply ACL 1 to a subset of the ports associated with virtual interface 1.

Syntax: [**no**] **ip access-group** *ACL ID* **in ethernet** *port* [**to** *port*]

The *ACL ID* parameter is the access list name or number.

Specify the port variable in *stack-unit/slotnum/portnum* format.

ACLs to filter ARP packets

NOTE
This feature is not applicable to outbound traffic.

You can use ACLs to filter ARP packets. Without this feature, ACLs cannot be used to permit or deny incoming ARP packets. Although an ARP packet contains an IP address just as an IP packet does, an ARP packet is not an IP packet; therefore, it is not subject to normal filtering provided by ACLs.

When a Brocade device receives an ARP request, the source MAC and IP addresses are stored in the device ARP table. A new record in the ARP table overwrites existing records that contain the same IP address. This behavior can cause a condition called "ARP hijacking", when two hosts with the same IP address try to send an ARP request to the Brocade device.

Normally ARP hijacking is not a problem because IP assignments are done dynamically; however, in some cases, ARP hijacking can occur, such as when a configuration allows a router interface to share the IP address of another router interface. Since multiple VLANs and the router interfaces that are associated with each of the VLANs share the same IP segment, it is possible for two hosts in two different VLANs to fight for the same IP address in that segment. ARP filtering using ACLs protects an IP host record in the ARP table from being overwritten by a hijacking host. Using ACLs to filter ARP requests checks the source IP address in the received ARP packet. Only packets with the permitted IP address will be allowed to be to be written in the ARP table; others are dropped.

Configuration considerations for filtering ARP packets

- This feature is available on devices running Layer 3 code. This filtering occurs on the management processor.
- The feature is available on physical interfaces and virtual routing interfaces. It is supported on the following physical interface types Ethernet and trunks.
- ACLs used to filter ARP packets on a virtual routing interface can be inherited from a previous interface if the virtual routing interface is defined as a follower virtual routing interface.

Configuring ACLs for ARP filtering

To implement the ACL ARP filtering feature, enter commands such as the following.

```
Brocade(config)# access-list 101 permit ip host 192.168.2.2 any
Brocade(config)# access-list 102 permit ip host 192.168.2.3 any
Brocade(config)# access-list 103 permit ip host 192.168.2.4 any
Brocade(config)# vlan 2
Brocade(config-vlan-2)# tag ethernet 1/1/1 to 1/1/2
Brocade(config-vlan-2)# router-interface ve 2
Brocade(config-vlan-2)# vlan 3
Brocade(config-vlan-3)# tag ethernet 1/1/1 to 1/1/2
Brocade(config-vlan-3)# router-interface ve 3
Brocade(config-vlan-3)# vlan 4
Brocade(config-vlan-4)# tag ethe 1/1/1 to 1/1/2
Brocade(config-vlan-4)# router-interface ve 4
Brocade(config-vlan-4)# interface ve 2
Brocade(config-ve-2)# ip access-group 101 in
Brocade(config-ve-2)# ip address 192.168.2.1/24
Brocade(config-ve-2)# ip use-ACL-on-arp 103
Brocade(config-ve-2)# exit
Brocade(config)# interface ve 3
Brocade(config-ve-3)# ip access-group 102 in
Brocade(config-ve-3)# ip follow ve 2
Brocade(config-ve-3)# ip use-ACL-on-arp
Brocade(config-ve-3)# exit
Brocade(config-vlan-4)# interface ve 4
Brocade(config-ve-4)# ip follow ve 2
Brocade(config-ve-4)# ip use-ACL-on-arp
Brocade(config-ve-4)# exit
```

Syntax: [**no**] **ip use-ACL-on-arp** [*access-list-number*]

When the **use-ACL-on-arp** command is configured, the ARP module checks the source IP address of the ARP request packets received on the interface. It then applies the specified ACL policies to the packet. Only the packet with the IP address that the ACL permits will be allowed to be to be written in the ARP table; those that are not permitted will be dropped.

The *access-list-number* parameter identifies the ID of the standard ACL that will be used to filter the packet. Only the source and destination IP addresses will be used to filter the ARP packet. You can do one of the following for *access-list-number*:

- Enter an ACL ID to explicitly specify the ACL to be used for filtering. In the example above, the line `Brocade(config-ve-2)# ip use-ACL-on-arp 103` specifies ACL 103 to be used as the filter.

- Allow the ACL ID to be inherited from the IP ACLs that have been defined for the device. In the example above, the line `Brocade(config-ve-4)# ip use-ACL-on-arp` allows the ACL to be inherited from IP ACL 101 because of the ip follow relationship between virtual routing interface 2 and virtual routing interface 4. Virtual routing interface 2 is configured with IP ACL 101; thus virtual routing interface 4 inherits IP ACL 101.

ARP requests will not be filtered by ACLs if one of the following conditions occur:

- If the ACL is to be inherited from an IP ACL, but there is no IP ACL defined.
- An ACL ID is specified for the use-ACL-on-arp command, but no IP address or “any any” filtering criteria have been defined under the ACL ID.

Displaying ACL filters for ARP

To determine which ACLs have been configured to filter ARP requests, enter a command such as the following.

```
Brocade(config)# show ACL-on-arp
Port  ACL ID Filter Count
1/1/2 103 10
1/1/3 102 23
1/1/4 101 12
```

Syntax: show ACL-on-arp [**ethernet** *port* | **loopback** [*num*] | **ve** [*num*]]

Specify the *port* variable in *slotnum/portnum* format.

If the *port* variable is not specified, all ports on the device that use ACLs for ARP filtering will be included in the display.

The Filter Count column shows how many ARP packets have been dropped on the interface since the last time the count was cleared.

Clearing the filter count

To clear the filter count for all interfaces on the device, enter a command such as the following.

```
Brocade(config)# clear ACL-on-arp
```

The above command resets the filter count on all interfaces in a device back to zero.

Syntax: clear ACL-on-arp

Filtering on IP precedence and ToS values

To configure an extended IP ACL that matches based on IP precedence, enter commands such as the following.

```
Brocade(config)# access-list 103 deny tcp 10.157.21.0/24 10.157.22.0/24
precedence internet
Brocade(config)# access-list 103 deny tcp 10.157.21.0/24 eq ftp 10.157.22.0/24
precedence 6
Brocade(config)# access-list 103 permit ip any any
```

The first entry in this ACL denies TCP traffic from the 10.157.21.x network to the 10.157.22.x network, if the traffic has the IP precedence option “internet” (equivalent to “6”).

The second entry denies all FTP traffic from the 10.157.21.x network to the 10.157.22.x network, if the traffic has the IP precedence value “6” (equivalent to “internet”).

The third entry permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL would deny all incoming or outgoing IP traffic on the ports to which you assign the ACL.

To configure an IP ACL that matches based on ToS, enter commands such as the following.

```
Brocade(config)# access-list 104 deny tcp 10.157.21.0/24 10.157.22.0/24 tos
normal
Brocade(config)# access-list 104 deny tcp 10.157.21.0/24 eq ftp 10.157.22.0/24
tos 13
Brocade(config)# access-list 104 permit ip any any
```

The first entry in this IP ACL denies TCP traffic from the 10.157.21.x network to the 10.157.22.x network, if the traffic has the IP ToS option “normal” (equivalent to “0”).

The second entry denies all FTP traffic from the 10.157.21.x network to the 10.157.22.x network, if the traffic has the IP ToS value “13” (equivalent to “max-throughput”, “min-delay”, and “min-monetary-cost”).

The third entry permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL would deny all incoming or outgoing IP traffic on the ports to which you assign the ACL.

TCP flags - edge port security

The edge port security feature works in combination with IP ACL rules and can be combined with other ACL functions (such as dscp-marking and traffic policies), giving you greater flexibility when designing ACLs.

For details about the edge port security feature, refer to “Using TCP Flags in combination with other ACL features” on page 61.

QoS options for IP ACLs

Quality of Service (QoS) options enable you to perform QoS for packets that match the ACLs. Using an ACL to perform QoS is an alternative to directly setting the internal forwarding priority based on incoming port, VLAN membership, and so on. (This method is described in *Brocade ICX 6650 Platform and Layer 2 Switching Configuration Guide*.)

The following QoS ACL options are supported:

- **dscp-cos-mapping** – By default, the Brocade device does the *802.1p* to *CoS* mapping.
- **dscp-marking** – Marks the DSCP value in the outgoing packet with the value you specify.
- **internal-priority-marking** and **802.1p-priority-marking** – Supported with the DSCP marking option, these commands assign traffic that matches the ACL to a hardware forwarding queue (**internal-priority-marking**), and re-mark the packets that match the ACL with the 802.1p priority (**802.1p-priority-marking**).

- **dscp-matching** – Matches on the packet DSCP value. This option does not change the packet forwarding priority through the device or mark the packet.
- **802.1p-priority-matching** – Inspects the 802.1p bit in the ACL that can be used with adaptive rate limiting.

NOTE
These QoS options are only available if a specific ICMP type is specified for the *icmp-type* parameter while configuring extended ACLS, and cannot be used with the **any-icmp-type** option. See "Extended numbered ACL syntax" on page 91 and "Extended named ACL syntax" on page 97for the syntax for configuring extended ACLs.

Configuration notes for QoS options on Brocade ICX 6650

These devices do not support marking and prioritization simultaneously with the same rule (and do not support DSCP CoS mapping at all). To achieve this, you need to create two separate rules. In other words, you can mark a rule with DSCP or 802.1p information, or you can prioritize a rule based on DSCP or 802.1p information. You can enable only one of the following ACL options per rule:

- 802.1p-priority-marking
- dscp-marking

For example, any one of the following commands is supported.

```
Brocade(config)#access-list 101 permit ip any any dscp-marking 43
```

or

```
Brocade(config)#access-list 101 permit ip any any 802.1p-priority-marking
```

Using an IP ACL to mark DSCP values (DSCP marking)

The **dscp-marking** option for extended ACLs allows you to configure an ACL that marks matching packets with a specified DSCP value. You also can use DSCP marking to assign traffic to a specific hardware forwarding queue (refer to "Using an ACL to change the forwarding queue" on page 117).

For example, the following commands configure an ACL that marks all IP packets with DSCP value 5. The ACL is then applied to incoming packets on interface 7. Consequently, all inbound packets on interface 7 are marked with the specified DSCP value.

```
Brocade(config)# access-list 120 permit ip any any dscp-marking 5
Brocade(config)# interface 1/1/7
Brocade(config-if-e10000-1/1/7)# ip access-group 120 in
```

Syntax: ...dscp-marking *dscp-value*

The **dscp-marking** *dscp-value* parameter maps a DSCP value to an internal forwarding priority. The DSCP value can be from 0-63.

Combined ACL for 802.1p marking

Brocade devices support a simple method for assigning an 802.1p priority value to packets without affecting the actual packet or the DSCP. In early IronWare software releases, users were required to provide DSCP-marking and DSCP-matching information in order to assign 802.1p priority values, which required the deployment of a 64-line ACL to match all possible DSCP values. Users were also required to configure an internal priority marking value. Now, users can easily specify 802.1p priority marking values directly, and change internal priority marking from *required* to o*ptional*.

NOTE
This feature is not applicable to outbound traffic.

On Brocade ICX 6650, if the user does not set a specific internal marking priority, the default value is the same as the 802.1-priority marking value:

Priority values range from 0 to 7.

Two new ACL parameters support this feature, one required for priority marking and one optional for internal priority marking. These parameters apply to IP, and TCP, and UDP.

NOTE
Brocade ICX 6650 does not allow setting 802.1p-priority-marking value different from the internal-priority-marking value. You can have both parameters configured in a single ACL rule if both values are the same.

For IP

```
Brocade(config)# access-list 104 per ip any any 802.1p-priority-marking 1
```

or the following command, which also assigns an optional internal-priority-marking value.

```
Brocade(config)# access-list 104 per ip any any 802.1p-priority-marking 1
internal-priority-marking 1
```

Syntax: **access-list** *num(100-199)* **permit ip any any 802.1p-priority-marking** *priority value (0-7)* [**internal-priority-marking** *value (0-7)*]

For TCP

```
Brocade(config)# access-list 105 per tcp any any 802.1p-priority-marking 1
```

or the following command, which also assigns an optional internal-priority-marking value.

```
Brocade(config)# access-list 105 per tcp any any 802.1p-priority-marking 1
internal-priority-marking 1
```

Syntax: **access-list** *num(100-199)* **permit tcp any any 802.1p-priority-marking** *priority value (0-7)* [**internal-priority-marking** *value (0-7)*]

For UDP

```
Brocade(config)# access-list 105 per udp any any 802.1p-priority-marking 1
```

or the following command, which also assigns an optional internal-priority-marking value.

```
Brocade(config)# access-list 105 per udp any any 802.1p-priority-marking 1
internal-priority-marking 1
```

Syntax: **access-list** *num(100-199)* **permit udp any any 802.1p-priority-marking** *priority value (0-7)* [**internal-priority-marking** *value (0-7)*]

In each of these examples, in the first command the internal-priority value is not specified, which means it maintains a default value of 1 (equal to that of the 802.1p value).

Using an ACL to change the forwarding queue

The **802.1p-priority-marking** *0 – 7* parameter re-marks the packets of the 802.1Q traffic that match the ACL with this new 802.1p priority, or marks the packets of the non-802.1Q traffic that match the ACL with this 802.1p priority, later at the outgoing 802.1Q interface.

The **internal-priority-marking** *0 – 7* parameter assigns traffic that matches the ACL to a specific hardware forwarding queue (qosp0 – qosp7>.

NOTE
The **internal-priority-marking** parameter overrides port-based priority settings.

In addition to changing the internal forwarding priority, if the outgoing interface is an 802.1Q interface, this parameter maps the specified priority to its equivalent 802.1p (CoS) priority and marks the packet with the new 802.1p priority. The complete CLI syntax for 802.1p priority marking and internal priority marking is shown in "Extended numbered ACL configuration" on page 90 and "Extended named ACL configuration" on page 96. The following shows the syntax specific to these features.

Syntax: **... dscp-marking** <*0 – 63*> **802.1p-priority-marking** <*0 – 7*> **internal-priority-marking** <*0 – 7*>]

DSCP matching

The **dscp-matching** option matches on the packet DSCP value. This option does not change the packet forwarding priority through the device or mark the packet.

To configure an ACL that matches on a packet with DSCP value 29, enter a command such as the following.

```
Brocade(config)# access-list 112 permit ip 10.1.1.0 0.0.0.255 10.2.2.x 0.0.0.255
dscp-matching 29
```

The complete CLI syntax for this feature is shown in "Extended numbered ACL configuration" on page 90 and "Extended named ACL configuration" on page 96. The following shows the syntax specific to this feature.

Syntax: **...dscp-matching** <*0 – 63*>

NOTE
For complete syntax information, refer to "Extended numbered ACL syntax" on page 91.

ACL-based rate limiting

ACL-based rate limiting provides the facility to limit the rate for IP traffic that matches the permit conditions in extended IP ACLs. This feature is available in the Layer 2 and Layer 3 code.

NOTE
Brocade devices support ACL-based rate limiting for inbound traffic. This feature is not supported for outbound traffic.

For more details, including configuration procedures, refer to Chapter 5, "ACL-based Rate Limiting".

ACL statistics

ACL statistics is a mechanism for counting the number of packets and the number of bytes per packet to which ACL filters are applied.

To see the configuration procedures for ACL statistics, refer to Chapter 5, "ACL-based Rate Limiting".

NOTE
The terms *ACL statistics* and *ACL counting* are used interchangeably in this guide and mean the same thing.

ACLs to control multicast features

You can use ACLs to control the following multicast features:

- Limit the number of multicast groups that are covered by a static rendezvous point (RP)
- Control which multicast groups for which candidate RPs sends advertisement messages to bootstrap routers
- Identify which multicast group packets will be forwarded or blocked on an interface

For configuration procedures, refer to *Brocade ICX 6650 IP Multicast Configuration Guide*.

Enabling and viewing hardware usage statistics for an ACL

The number of configured ACL rules can affect the rate at which hardware resources are used. You can use the **show access-list hw-usage on** command to enable hardware usage statistics, followed by the **show access-list** *access-list-id* command to determine the hardware usage for an ACL. To gain more hardware resources, you can modify the ACL rules so that it uses less hardware resource.

NOTE
The hardware usage statistics will only be shown for IPv4 ACLs.

To enable and view hardware usage statistics, enter commands such as the following:

```
Brocade# show access-list hw-usage on
Brocade# show access-list 100
Extended IP access list 100 (hw usage : 2)
deny ip any any (hw usage : 1
```

The first command enables hardware usage statistics, and the second command displays the hardware usage for IP access list 100.

Syntax: show access-list hw-usage on | off

Syntax: show access-list *access-list-id* | **all**

By default, hardware usage statistics are disabled. To disable hardware usage statistics after is has been enabled, use the **show access-list hw-usage off** command.

The *access-list-id* variable is a valid ACL name or number.

Displaying ACL information

To display the number of entries used by each ACL, enter the following command.

```
Brocade# show ip access-lists

Extended IP access list 100: 1 entry
deny ip any any
```

Syntax: show access-list *ACL-num* | *ACL-name* | **all**

The Rule cam use field lists the number of CAM entries used by the ACL or entry. The number of CAM entries listed for the ACL itself is the total of the CAM entries used by the ACL entries.

For flow-based ACLs, the Total flows and Flows fields list the number of Layer 4 session table flows in use for the ACL.

The Total packets and Packets fields apply only to flow-based ACLs.

Troubleshooting ACLs

Use the following methods to troubleshoot access control lists (ACLs):

- To display the number of Layer 4 CAM entries being used by each ACL, enter the **show access-list** *ACL-num* | *ACL-name* | **all** command. Refer to "Displaying ACL information" on page 119.
- To determine whether the issue is specific to fragmentation, remove the Layer 4 information (TCP or UDP application ports) from the ACL, then reapply the ACL.

If you are using another feature that requires ACLs, either use the same ACL entries for filtering and for the other feature, or change to flow-based ACLs.

Policy Based Routing

Policy-Based Routing (PBR) allows you to use ACLs and route maps to selectively modify and route IP packets in hardware. The ACLs classify the traffic. 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. Using standard ACLs with PBR, you can route IP packets based on their source IP address. With extended ACLs, you can route IP packets based on all of the clauses in the extended ACL.

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

- Select the next-hop gateway.
- Send the packet to the null interface (null0).

When a PBR policy has multiple next hops to a destination, PBR selects the first live next hop specified in the policy that is up. If none of the policy's direct routes or next hops are available, the packet is routed in the normal way.

Configuration considerations for policy-based routing

- PBR is supported in the full Layer 3 code only.
- PBR is not supported together with ACLs on the same port.
- Global PBR is not supported when IP Follow is configured on an interface.
- Global PBR is not supported with per-port-per-VLAN ACLs.
- A PBR policy on an interface takes precedence over a global PBR policy.
- You cannot apply PBR on a port if that port already has ACLs, ACL-based rate limiting, DSCP-based QoS, MAC address filtering.
- The number of route maps that you can define is limited by the available system memory, which is determined by the system configuration and how much memory other features use. When a route map is used in a PBR policy, the PBR policy uses up to six instances of a route map, up to five ACLs in a matching policy of each route map instance, and up to six next hops in a set policy of each route map instance. Note that the CLI will allow you configure more than six next hops in a route map; however, the extra next hops will not be placed in the PBR database. The route map could be used by other features like BGP or OSPF, which may use more than six next hops.
- ACLs with the **log** option configured should not be used for PBR purposes.
- PBR ignores explicit or implicit **deny ip any any** ACL entries, to ensure that for route maps that use multiple ACLs, the traffic is compared to all the ACLs. PBR also ignores any deny clauses in an ACL. Traffic that matches a deny clause is routed normally using Layer 3 paths.
- PBR always selects the first next hop from the next hop list that is up. If a PBR policy's next hop goes down, the policy uses another next hop if available. If no next hops are available, the device routes the traffic in the normal way.
- PBR is not supported for fragmented packets. If the PBR ACL filters on Layer 4 information like TCP/UDP ports, fragmented packed are routed normally.
- You can change route maps or ACL definitions dynamically and do not need to rebind the PBR policy to an interface.

Configuring a PBR policy

To configure PBR, you define the policies using IP ACLs and route maps, then enable PBR globally or on individual interfaces. The device programs the ACLs into the packet processor on the interfaces and routes traffic that matches the ACLs according to the instructions in the route maps.

To configure a PBR policy:

- Configure ACLs that contain the source IP addresses for the IP traffic you want to route using PBR.

- Configure a route map that matches on the ACLs and sets the route information.
- Apply the route map to an interface.

Configuring the ACLs

PBR uses route maps to change the routing attributes in IP traffic. This section shows an example of how to configure a standard ACL to identify the source subnet for IP traffic.

To configure a standard ACL to identify a source subnet, enter a command such as the following.

```
Brocade(config)# access-list 99 permit 10.157.23.0 0.0.0.255
```

The command in this example configures a standard ACL that permits traffic from subnet 10.157.23.0/24. After you configure a route map that matches based on this ACL, the software uses the route map to set route attributes for the traffic, thus enforcing PBR.

NOTE
Do not use an access group to apply the ACL to an interface. Instead, use a route map to apply the ACL globally or to individual interfaces for PBR, as shown in the following sections.

Syntax: [**no**]**access-list** *num* **deny** | **permit** *source-ip* | *hostname wildcard*

or

Syntax: [**no**]**access-list** *num* **deny** | **permit** *source-ip/mask-bits* |*hostname*

Syntax: [**no**]**access-list** *num* **deny** | **permit host** *source-ip* | *hostname*

Syntax: [**no**]**access-list** *num* **deny** | **permit any**

The *num* parameter is the access list number and can be from 1–99.

The **deny** | **permit** parameter indicates whether packets that match a policy in the access list are denied (dropped) or permitted (forwarded).

NOTE
If you are configuring the ACL for use in a route map, always specify **permit**. Otherwise, the Brocade device will ignore deny clauses and packets that match deny clauses are routed normally.

The *source-ip* parameter specifies the source IP address. Alternatively, you can specify the host name.

NOTE
To specify the host name instead of the IP address, the host name must be configured using the DNS resolver on the Brocade device. To configure the DNS resolver name, use the **ip dns server-address**... command at the global CONFIG level of the CLI.

The *wildcard* parameter specifies the mask value to compare against the host address specified by the *source-ip* parameter. The *wildcard* is in dotted-decimal notation (IP address format). It is a four-part value, where each part is 8 bits (one byte) separated by dots, and each bit is a one or a zero. Each part is a number ranging from 0 to 255, for example 0.0.0.255. Zeros in the mask mean the packet source address must match the *source-ip*. Ones mean any value matches. For example, the *source-ip* and *wildcard* values 10.157.22.26 0.0.0.255 mean that all hosts in the Class A subnet 10.157.22.x match the policy.

If you prefer to specify the wildcard (mask value) in CIDR format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of "10.157.22.26 0.0.0.255" as "10.157.22.26/24". The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into zeros. For example, if you specify 10.157.22.26/24 or 10.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 10.157.22.0/24 (if you have enabled display of subnet lengths) or 10.157.22.0 0.0.0.255 in the startup-config file.

If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in "*/mask-bits*" format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

NOTE
If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show ip access-list** command.

The **host** *source-ip* | *hostname* parameter lets you specify a host IP address or name. When you use this parameter, you do not need to specify the mask. A mask of all zeros (0.0.0.0) is implied.

The **any** parameter configures the policy to match on all host addresses.

NOTE
Do not use the **log** option in ACLs that will be used for PBR.

Configuring the route map

After you configure the ACLs, you can configure a PBR route map that matches based on the ACLs and sets routing information in the IP traffic.

NOTE
The match and set statements described in this section are the only route-map statements supported for PBR. Other route-map statements described in the documentation apply only to the protocols with which they are described.

To configure a PBR route map, enter commands such as the following.

```
Brocade(config)# route-map test-route permit 99
Brocade(config-routemap test-route)# match ip address 99
Brocade(config-routemap test-route)# set ip next-hop 192.168.2.1
Brocade(config-routemap test-route)# exit
```

The commands in this example configure an entry in a route map named "test-route". The **match** statement matches on IP information in ACL 99. The **set** statement changes the next-hop IP address for packets that match to 192.168.2.1.

Syntax: [**no**]**route-map** *map-name* **permit** | **deny** *num*

The *map-name* is a string of characters that names the map. Map names can be up to 32 characters in length. You can define an unlimited number of route maps on the Brocade device, as long as system memory is available.

The **permit** | **deny** parameter specifies the action the Brocade device will take if a route matches a match statement:

- If you specify **deny**, the Brocade device does not apply a PBR policy to packets that match the ACLs in a match clause. Those packets are routed normally,
- If you specify **permit**, the Brocade device applies the match and set statements associated with this route map instance.

The *num* parameter specifies the instance of the route map you are defining. Routes are compared to the instances in ascending numerical order. For example, a route is compared to instance 1, then instance 2, and so on.

PBR uses up to six route map instances for comparison and ignores the rest.

Syntax: [**no**] **match ip address** *ACL-num-or-name*

The *ACL-num* parameter specifies a standard or extended ACL number or name.

Syntax: [**no**] **set ip next hop** *ip-addr*

This command sets the next-hop IP address for traffic that matches a match statement in the route map.

Syntax: [**no**] **set interface null0**

This command sends the traffic to the null0 interface, which is the same as dropping the traffic.

Enabling PBR

After you configure the ACLs and route map entries, you can enable PBR globally, on individual interfaces, or both as described in this section. To enable PBR, you apply a route map you have configured for PBR globally or locally.

Enabling PBR globally

To enable PBR globally, enter a command such as the following at the global CONFIG level.

```
Brocade(config)# ip policy route-map test-route
```

This command applies a route map named “test-route” to all interfaces on the device for PBR.

Syntax: ip policy route-map *map-name*

Enabling PBR locally

To enable PBR locally, enter commands such as the following.

```
Brocade(config)# interface ve 1
Brocade(config-vif-1)# ip policy route-map test-route
```

The commands in this example change the CLI to the Interface level for virtual interface 1, then apply the “test-route” route map to the interface. You can apply a PBR route map to Ethernet ports or virtual interfaces.

Syntax: ip policy route-map *map-name*

Enter the name of the route map you want to use for the route-map *map-name* parameter.

Configuration examples for PBR

This section presents configuration examples for configuring and applying a PBR policy.

Basic example of PBR

The following commands configure and apply a PBR policy that routes HTTP traffic received on virtual routing interface 1 from the 10.10.10.x/24 network to 10.5.5.x/24 through next-hop IP address 10.1.1.1/24 or, if 10.1.1.x is unavailable, through 10.2.2.1/24.

```
Brocade(config)# access-list 101 permit tcp 10.10.10.0 0.0.0.255 eq http 10.5.5.0
0.0.0.255
Brocade(config)# route-map net10web permit 101
Brocade(config-routemap net10web)# match ip address 101
Brocade(config-routemap net10web)# set ip next-hop 10.1.1.1
Brocade(config-routemap net10web)# set ip next-hop 10.2.2.2
Brocade(config-routemap net10web)# exit
Brocade(config)# vlan 10
Brocade(config-vlan-10)# tagged ethernet 1/1/1 to 1/1/4
Brocade(config-vlan-10)# router-interface ve 1
Brocade(config)# interface ve 1
Brocade(config-vif-1)# ip policy route-map net10web
```

Syntax: [**no**] **route-map** *map-name* **permit** | **deny** *num*

Syntax: [**no**] **set ip next hop** *ip-addr*

This command sets the next-hop IP address for traffic that matches a match statement in the route map.

Setting the next hop

The following commands configure the Brocade device to apply PBR to traffic from IP subnets 10.157.23.x, 10.157.24.x, and 10.157.25.x. In this example, route maps specify the next-hop gateway for packets from each of these subnets:

- Packets from 10.157.23.x are sent to 192.168.2.1.
- Packets from 10.157.24.x are sent to 192.168.2.2.
- Packets from 10.157.25.x are sent to 192.168.2.3.

The following commands configure three standard ACLs. Each ACL contains one of the ACLs listed above. Make sure you specify **permit** instead of deny in the ACLs, so that the Brocade device permits the traffic that matches the ACLs to be further evaluated by the route map. If you specify **deny**, the traffic that matches the deny statements are routed normally. Notice that these ACLs specify **any** for the destination address.

```
Brocade(config)# access-list 50 permit 10.157.23.0 0.0.0.255
Brocade(config)# access-list 51 permit 10.157.24.0 0.0.0.255
Brocade(config)# access-list 52 permit 19.157.25.0 0.0.0.255
```

The following commands configure three entries in a route map called “test-route”. The first entry (permit 50) matches on the IP address information in ACL 50 above. For IP traffic from subnet 10.157.23.0/24, this route map entry sets the next-hop IP address to 192.168.2.1.

```
Brocade(config)# route-map test-route permit 50
Brocade(config-routemap test-route)# match ip address 50
Brocade(config-routemap test-route)# set ip next-hop 192.168.2.1
Brocade(config-routemap test-route)# exit
```

The following commands configure the second entry in the route map. This entry (permit 51) matches on the IP address information in ACL 51 above. For IP traffic from subnet 10.157.24.0/24, this route map entry sets the next-hop IP address to 192.168.2.2.

```
Brocade(config)# route-map test-route permit 51
Brocade(config-routemap test-route)# match ip address 51
Brocade(config-routemap test-route)# set ip next-hop 192.168.2.2
Brocade(config-routemap test-route)# exit
```

The following commands configure the third entry in the test-route route map. This entry (permit 52) matches on the IP address information in ACL 52 above. For IP traffic from subnet 10.157.25.0/24, this route map entry sets the next-hop IP address to 192.168.2.3.

```
Brocade(config)# route-map test-route permit 52
Brocade(config-routemap test-route)# match ip address 52
Brocade(config-routemap test-route)# set ip next-hop 192.168.2.3
Brocade(config-routemap test-route)# exit
```

The following command enables PBR by globally applying the test-route route map to all interfaces.

```
Brocade(config)# ip policy route-map test-route
```

Alternatively, you can enable PBR on specific interfaces, as shown in the following example. The commands in this example configure IP addresses in the three source subnets identified in ACLs 50, 51, and 52, then apply route map test-route to the interface.

```
Brocade(config)# interface ve 1
Brocade(config-vif-1)# ip address 10.157.23.1/24
Brocade(config-vif-1)# ip address 10.157.24.1/24
Brocade(config-vif-1)# ip address 10.157.25.1/24
Brocade(config-vif-1)# ip policy route-map test-route
```

Setting the output interface to the null interface

The following commands configure a PBR policy to send all traffic from 192.168.1.204/32 to the null interface, thus dropping the traffic instead of forwarding it.

```
Brocade(config)# access-list 56 permit 10.168.1.204 0.0.0.0
```

The following commands configure an entry in a route map called “file-13”. The first entry (permit 56) matches on the IP address information in ACL 56 above. For IP traffic from the host 10.168.1.204/32, this route map entry sends the traffic to the null interface instead of forwarding it, thus sparing the rest of the network the unwanted traffic.

```
Brocade(config)# route-map file-13 permit 56
Brocade(config-routemap file-13)# match ip address 56
Brocade(config-routemap file-13)# set interface null0
Brocade(config-routemap file-13)# exit
```

The following command enables PBR by globally applying the route map to all interfaces.

```
Brocade(config)# ip policy route-map file-13
```

Alternatively, you can enable PBR on specific interfaces, as shown in the following example. The commands in this example configure IP addresses in the source subnet identified in ACL 56, then apply route map file-13 to the interface.

```
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# ip address 192.168.1.204/32
Brocade(config-if-e10000-1/3/1)# ip policy route-map file-13
```

Trunk formation with PBR policy

When a trunk is formed, the PBR policy on the primary port applies to all the secondary ports. If a different PBR policy exists on a secondary port at the time of a trunk formation, that policy is overridden by the PBR policy on the primary port. If the primary port does not have a PBR policy, then the secondary ports will not have a PBR policy.

When a trunk is removed, the PBR policy that was applied to the trunk interface is unbound (removed) from former secondary ports. If global PBR is configured, the secondary ports adhere to the global PBR; otherwise, no PBR policy is bound to former secondary ports.

Chapter

4

IPv6 ACLs

Table 17 lists the IPv6 Access Control Lists (ACL) features supported on Brocade ICX 6650. These features are supported in Brocade ICX 6650 that can be configured as an IPv6 host in an IPv6 network, and in devices that support IPv6 routing. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 17 Supported IPv6 ACL features

Feature	Brocade ICX 6650
IPv6 ACLs	Yes
Applying an IPv6 ACL to an interface	Yes
IPv6 ACL comment text	Yes
IPv6 ACL logging of denied packets	Yes

This chapter describes how ACLs are implemented and configured on a Brocade device.

IPv6 ACL overview

Brocade devices support IPv6 Access Control Lists (ACLs) for inbound traffic filtering, as detailed in Table 17. You can configure up to 100 IPv6 ACLs and, by default, up to a system-wide maximum of 8192 ACL rules.

An IPv6 ACL is composed of one or more conditional statements that pose an action (permit or deny) if a packet matches a specified source or destination prefix. For Brocade ICX 6650, there can be up to 2045 total hardware entries. Most IPv6 ACL rules will need 2 hardware entries, and some more than 2, per port region, including IPv6, IPv4, MAC address filters, and default statements. When the maximum number of ACL rules allowed per port region is reached, an error message will display on the console.

The last statement in each IPv6 ACL is an implicit deny statement for all packets that do not match the previous statements in the ACL.

You can configure an IPv6 ACL on a global basis, then apply it to the incoming IPv6 packets on specified interfaces. You can apply only one IPv6 ACL to an interface. When an interface receives an IPv6 packet, it applies the statements within the ACL in their order of appearance to the packet. As soon as a match occurs, the Brocade device takes the specified action (permit or deny the packet) and stops further comparison for that packet.

IPv6 ACLs are supported on:

- Gbps Ethernet ports
- 10 Gbps Ethernet ports
- Trunk groups
- Virtual routing interfaces

NOTE
IPv6 ACLs are supported on inbound traffic and are implemented in hardware, making it possible for the Brocade device to filter traffic at line-rate speed on 10 Gbps interfaces.

IPv6 ACL traffic filtering criteria

The Brocade implementation of IPv6 ACLs enable traffic filtering based on the following information:

- IPv6 protocol
- Source IPv6 address
- Destination IPv6 address
- IPv6 message type
- Source TCP or UDP port (if the IPv6 protocol is TCP or UDP)
- Destination TCP or UDP port (if the IPv6 protocol is TCP or UDP)

IPv6 protocol names and numbers

The IPv6 protocol can be one of the following well-known names or any IPv6 protocol number from 0 through 255:

- Authentication Header (AHP)
- Encapsulating Security Payload (ESP)
- Internet Control Message Protocol (ICMP)
- Internet Protocol Version 6 (IPv6)
- Stream Control Transmission Protocol (SCTP)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

NOTE
TCP and UDP filters will be matched only if they are listed as the first option in the extension header.

For TCP and UDP, you also can specify a comparison operator and port name or number. For example, you can configure a policy to block web access to a specific website by denying all TCP port 80 (HTTP) packets from a specified source IPv6 address to the website IPv6 address.

IPv6 ACLs also provide support for filtering packets based on DSCP.

IPv6 ACL configuration notes

- IPv4 ACLs that filter based on VLAN membership or VE port membership (ACL-per-port-per-VLAN), are supported together with IPv6 ACLs on the same device, as long as they are not bound to the same port or virtual interface.
- IPv4 source guard and IPv6 ACLs are supported together on the same device, as long as they are not configured on the same port or virtual Interface.
- IPv6 ACLs do not support ACL filtering based on VLAN membership or VE port membership.

- IPv6 ACLs cannot be used with GRE
- IPv6 ACLs cannot be employed to implement a user-based ACL scheme
- If an IPv6 ACL has the implicit **deny** condition, make sure it also **permits** the IPv6 link-local address, in addition to the global unicast address. Otherwise, routing protocols such as OSPF will not work. To view the link-local address, use the **show ipv6 interface** command.
- IPv6 must be enabled on the interface before an ACL can be applied to it. If IPv6 is not enabled on the interface, the system will display the following error message.

  ```
  Brocade(config-if-e10000-1/1/7)# ipv6 traffic-filter netw in
  Error:  IPv6 is not enabled for interface 1/1/7
  ```

  To enable IPv6 on an interface, enter **ipv6 enable** at the Interface level of the CLI, or assign an IPv6 address to the interface as described in *Brocade ICX 6650 Administration Guide* and further discussed in *Brocade ICX 6650 Security Configuration Guide*.
- You cannot disable IPv6 on an interface to which an ACL is bound. Attempting to do so will cause the system to return the following error message.

  ```
  Brocade(config-if-e10000-1/1/7)# no ipv6 enable
  Error:  Port 7 has IPv6 ACL configured.  Cannot disable IPv6
  ```

  To disable IPv6, first remove the ACL from the interface.
- For notes on applying IPv6 ACLs to trunk ports, see "Applying an IPv6 ACL to a trunk group" on page 138.
- For notes on applying IPv6 ACLs to virtual ports, see "Applying an IPv6 ACL to a virtual interface in a protocol-based or subnet-based VLAN" on page 138.

Configuring an IPv6 ACL

Follow the steps given below to configure an IPv6 ACL.

1. Create the ACL.
2. Enable IPv6 on the interface to which the ACL will be applied.
3. Apply the ACL to the interface.

Example IPv6 configurations

To configure an access list that blocks all Telnet traffic received on port 1/1/1 from IPv6 host 2001:db8:e0bb::2, enter the following commands.

```
Brocade(config)# ipv6 access-list fdry
Brocade(config-ipv6-access-list-fdry)# deny tcp host 2001:db8:e0bb::2 any eq
telnet
Brocade(config-ipv6-access-list-fdry)# permit ipv6 any any
Brocade(config-ipv6-access-list-fdry)# exit
Brocade(config)# interface ethernet 1/1/1
Brocade(config-if-10000-1/1/1)# ipv6 enable
Brocade(config-if-e10000-1/1/1)# ipv6 traffic-filter fdry in
Brocade(config)# write memory
```

The following is another example of commands for configuring an ACL and applying it to an interface.

```
Brocade(config)# ipv6 access-list netw
Brocade(config-ipv6-access-list-netw)# permit icmp 2001:db8:e0bb::/64
2001:db8::/64
Brocade(config-ipv6-access-list-netw)# deny ipv6 host 2001:db8:e0ac::2 host
2000:2383:e0aa:0::24
Brocade(config-ipv6-access-list-netw)# deny udp any any
Brocade(config-ipv6-access-list-netw)# permit ipv6 any any
```

The first condition permits ICMP traffic from hosts in the 2001:db8:e0bb::x network to hosts in the 2001:db8::x network.

The second condition denies all IPv6 traffic from host 2001:db8:e0ac::2 to host 2001:db8:e0aa:0::24.

The third condition denies all UDP traffic.

The fourth condition permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL would deny all incoming IPv6 traffic on the ports to which you assigned the ACL.

The following commands apply the ACL "netw" to the incoming traffic on port 1/1/2 and to the incoming traffic on port 1/3/1.

```
Brocade(config)# interface ethernet 1/1/2
Brocade(config-if-e10000-1/1/2)# ipv6 enable
Brocade(config-if-e10000-1/1/2)# ipv6 traffic-filter netw in
Brocade(config-if-e10000-1/1/2)# exit
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# ipv6 enable
Brocade(config-if-e10000-1/3/1)# ipv6 traffic-filter netw in
Brocade(config)# write memory
```

Here is another example.

```
Brocade(config)# ipv6 access-list nextone
Brocade(config-ipv6-access-list rtr)# deny tcp 2001:db8:21::/24
2001:db8:22::/24
Brocade(config-ipv6-access-list rtr)# deny udp any range 5 6 2001:db8:22::/24
Brocade(config-ipv6-access-list rtr)# permit ipv6 any any
Brocade(config-ipv6-access-list rtr)# write memory
```

The first condition in this ACL denies TCP traffic from the 2001:db8:21::x network to the 2001:db8:22::x network.

The next condition denies UDP packets from any source with source UDP port in ranges 5 to 6 and whose destination is to the 2001:db8:22::/24 network.

The third condition permits all packets containing source and destination addresses that are not explicitly denied by the first two. Without this entry, the ACL would deny all incoming IPv6 traffic on the ports to which you assign the ACL.

A **show running-config** command displays the following.

```
 Brocade(config)# show running-config
ipv6 access-list rtr
 deny tcp 2001:db8:21::/24 2001:db8:22::/24
 deny udp any range rje 6 2001:db8:22::/24
 permit ipv6 any any
```

A **show ipv6 access-list** command displays the following.

```
 Brocade(config)# show ipv6 access-list rtr
ipv6 access-list rtr: 3 entries
 10: deny tcp 2001:db8:21::/24 2001:db8:22::/24
 20: deny udp any range rje 6 2001:db8:22::/24
 30: permit ipv6 any any
```

The following commands apply the ACL “rtr” to the incoming traffic on ports 1/2/1 and 1/2/2.

```
Brocade(config)# interface ethernet 1/2/1
Brocade(config-if-e10000-1/2/1)# ipv6 enable
Brocade(config-if-e10000-1/2/1)# ipv6 traffic-filter rtr in
Brocade(config-if-e10000-1/2/1)# exit
Brocade(config)# interface ethernet 1/2/2
Brocade(config-if-e10000-1/2/2)# ipv6 enable
Brocade(config-if-e10000-1/2/2)# ipv6 traffic-filter rtr in
Brocade(config)# write memory
```

Default and implicit IPv6 ACL action

The default action when no IPv6 ACLs are configured on an interface is to permit all IPv6 traffic. However, once you configure an IPv6 ACL and apply it to an interface, the default action for that interface is to deny all IPv6 traffic that is not explicitly permitted on the interface.

- If you want to tightly control access, configure ACLs consisting of permit entries for the access you want to permit. The ACLs implicitly deny all other access.
- If you want to secure access in environments with many users, you might want to configure ACLs that consist of explicit deny entries, then add an entry to permit all access to the end of each ACL. The permit entry permits packets that are not denied by the deny entries.

Every IPv6 ACL has the following implicit conditions as its last match conditions.

- **permit icmp any any nd-na** – Allows ICMP neighbor discovery acknowledgements.
- **permit icmp any any nd-ns** – Allows ICMP neighbor discovery solicitations.
- **deny ipv6 any any** – Denies IPv6 traffic. You must enter a **permit ipv6 any any** as the last statement in the access-list if you want to permit IPv6 traffic that were not denied by the previous statements.

NOTE
If an IPv6 ACL has the implicit **deny** condition, make sure it also **permits** the IPv6 link-local address, in addition to the global unicast address. Otherwise, routing protocols such as OSPF will not work. To view the link-local address, use the **show ipv6 interface** command.

The conditions are applied in the order shown above, with **deny ipv6 any any** as the last condition applied.

For example, if you want to deny ICMP neighbor discovery acknowledgement, then permit any remaining IPv6 traffic, enter commands such as the following.

```
Brocade(config)# ipv6 access-list netw
Brocade(config-ipv6-access-list-netw)# permit icmp 2001:db8:e0bb::/64
2001:3782::/64
Brocade(config-ipv6-access-list-netw)# deny icmp any any nd-na
Brocade(config-ipv6-access-list-netw)# permit ipv6 any any
```

The first permit statement permits ICMP traffic from hosts in the 2001:db8:e0bb::x network to hosts in the 2001:db8::x network.

The deny statement denies ICMP neighbor discovery acknowledgement.

The last entry permits all packets that are not explicitly denied by the other entries. Without this entry, the ACL will deny all incoming IPv6 traffic on the ports to which you assigned the ACL.

Furthermore, if you add the statement **deny icmp any any** in the access list, then all neighbor discovery messages will be denied. You must explicitly enter the **permit icmp any any nd-na** and **permit icmp any any nd-ns** statements just before the **deny icmp** statement if you want the ACLs to permit neighbor discovery as in the example below.

```
Brocade(config)# ipv6 access-list netw
Brocade(config-ipv6-access-list-netw)# permit icmp 2001:db8:e0bb::/64
2001:3782::/64
Brocade(config-ipv6-access-list-netw)# permit icmp any any nd-na
Brocade(config-ipv6-access-list-netw)# permit icmp any any nd-ns
Brocade(config-ipv6-access-list-netw)# deny icmp any any
Brocade(config-ipv6-access-list-netw)# permit ipv6 any any
```

Creating an IPv6 ACL

Before an IPv6 ACL can be applied to an interface, it must first be created, and then IPv6 must be enabled on that interface.

To create an IPv6 ACL, enter commands such as the following:

```
Brocade(config)# ipv6 access-list fdry
Brocade(config-ipv6-access-list-fdry)# deny tcp host 2001:db8:e0bb::2 any eq
telnet
Brocade(config-ipv6-access-list-fdry)# permit ipv6 any any
Brocade(config-ipv6-access-list-fdry)# exit
```

This creates an access list that blocks all Telnet traffic from IPv6 host 2001:db8:e0bb::2.

Syntax for creating an IPv6 ACL

- **ipv6-operator dscp**
- **ipv6-operator fragments** when any protocol is specified. The option "fragments" can be specified only when "permit/deny ipv6" is specified. If you specify "tcp" or any other protocol instead of "ipv6" the keyword, "fragments" cannot be used.
- **ipv6-operator routing** when any protocol is specified. (Same limitation as for **ipv6-operator fragments**)

When creating ACLs, use the appropriate syntax below for the protocol you are filtering.

For IPv6 and supported protocols other than ICMP, TCP, or UDP

Syntax: [**no**] **ipv6 access-list** *ACL-name*

Syntax: **permit** | **deny** *protocol*
ipv6-source-prefix/prefix-length | **any** | **host** *source-ipv6_address*
ipv6-destination-prefix/prefix-length | **any** | **host** *ipv6-destination-address*
[**ipv6-operator** [*value*]]
[**802.1p-priority-matching** *number*]
[**dscp-marking** *number* **802.1p-priority-marking** *number* **internal-priority-marking** *number*]
| [**dscp-marking** *dscp-value* **dscp-cos-mapping**]

For ICMP

Syntax: [**no**] **ipv6 access-list** *ACL name*

Syntax: **permit** | **deny icmp** *ipv6-source-prefix/prefix-length* | **any** | **host** *source-ipv6_address*
ipv6-destination-prefix/prefix-length | **any** | **host** *ipv6-destination-address*
[**ipv6-operator** [*value*]]
[[*icmp-type*][*icmp-code*]] | [*icmp-message*]
[**dscp-marking** *number*]

For TCP

Syntax: [**no**] **ipv6 access-list** *ACL-name*

Syntax: **permit** | **deny** *tcp*
ipv6-source-prefix/prefix-length | **any** | **host** *source-ipv6_address* [**tcp-udp-operator** [**source-port-number**]]
ipv6-destination-prefix/prefix-length | **any** | **host** *ipv6-destination-address*
[**tcp-udp-operator** [**destination-port- number**]]
[**ipv6-operator** [*value*]]
[**802.1p-priority-matching** *number*]
[**dscp-marking** *number* **802.1p-priority-marking** *number* **internal-priority-marking** *number*]

For UDP

Syntax: [**no**] **ipv6 access-list** *ACL-name*

Syntax: **permit** | **deny** *udp*
ipv6-source-prefix/prefix-length | **any** | **host** *source-ipv6_address* [**tcp-udp-operator** [**source port number**]]
ipv6-destination-prefix/prefix-length | **any** | **host** *ipv6-destination-address*
[**tcp-udp-operator** [**destination port number**]]
[**ipv6-operator** [*value*]]
[**802.1p-priority-matching** *number*]
[**dscp-marking** *number* **802.1p-priority-marking** *number* **internal-priority-marking** *number*]

Table 18 lists the syntax elements.

TABLE 18 Syntax descriptions

IPv6 ACL arguments	Description
ipv6 access-list *ACL-name*	Enables the IPv6 configuration level and defines the name of the IPv6 ACL. The *ACL-name* can contain up to 199 characters and numbers, but cannot begin with a number and cannot contain any spaces or quotation marks.
permit	The ACL will permit (forward) packets that match a policy in the access list.
deny	The ACL will deny (drop) packets that match a policy in the access list.
icmp	Indicates the you are filtering ICMP packets.
protocol	The type of IPv6 packet you are filtering. You can specify a well-known name for some protocols whose number is less than 255. For other protocols, you must enter the number. Enter "?" instead of a protocol to list the well-known names recognized by the CLI. IPv6 protocols include AHP – Authentication Header ESP – Encapsulating Security Payload IPv6 – Internet Protocol version 6 SCTP – Stream Control Transmission Protocol
ipv6-source-prefix/prefix-length	The *ipv6-source-prefix/prefix-length* parameter specify a source prefix and prefix length that a packet must match for the specified action (deny or permit) to occur. You must specify the *ipv6-source-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.
ipv6-destination-prefix/prefix-length	The *ipv6-destination-prefix/prefix-length* parameter specify a destination prefix and prefix length that a packet must match for the specified action (deny or permit) to occur. You must specify the *ipv6-destination-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter
any	When specified instead of the *ipv6-source-prefix/prefix-length* or *ipv6-destination-prefix/prefix-length* parameters, matches any IPv6 prefix and is equivalent to the IPv6 prefix::/0.
host	Allows you specify a host IPv6 address. When you use this parameter, you do not need to specify the prefix length. A prefix length of all128 is implied.
icmp-type	ICMP packets can be filtered by ICMP message type. The type is a number from 0 to 255.
icmp code	ICMP packets, which are filtered by ICMP message type can also be filtered by the ICMP message code. The code is a number from 0 to 255,
icmp-message	ICMP packets are filtered by ICMP messages. Refer to "ICMP message configurations" on page 136 for a list of ICMP message types.
tcp	Indicates the you are filtering TCP packets.
udp	Indicates the you are filtering UDP packets.

TABLE 18 Syntax descriptions (Continued)

IPv6 ACL arguments	Description
ipv6-source-prefix/prefix-length	The *ipv6-source-prefix/prefix-length* parameter specify a source prefix and prefix length that a packet must match for the specified action (deny or permit) to occur. You must specify the *ipv6-source-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.
ipv6-destination-prefix/prefix-length	The *ipv6-destination-prefix/prefix-length* parameter specify a destination prefix and prefix length that a packet must match for the specified action (deny or permit) to occur. You must specify the *ipv6-destination-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter
any	When specified instead of the *ipv6-source-prefix/prefix-length* or *ipv6-destination-prefix/prefix-length* parameters, matches any IPv6 prefix and is equivalent to the IPv6 prefix::/0.
host	Allows you specify a host IPv6 address. When you use this parameter, you do not need to specify the prefix length. A prefix length of all128 is implied.
tcp-udp-operator	The *tcp-udp-operator* parameter can be one of the following: • **eq** – The policy applies to the TCP or UDP port name or number you enter after **eq**. • **gt** – The policy applies to TCP or UDP port numbers greater than the port number or the numeric equivalent of the port name you enter after **gt**. Enter "?" to list the port names. • **lt** – The policy applies to TCP or UDP port numbers that are less than the port number or the numeric equivalent of the port name you enter after **lt**. • **neq** – The policy applies to all TCP or UDP port numbers except the port number or port name you enter after **neq**. • **range** – The policy applies to all TCP port numbers that are between the first TCP or UDP port name or number and the second one you enter following the **range** parameter. The range includes the port names or numbers you enter. For example, to apply the policy to all ports between and including 23 (Telnet) and 53 (DNS), enter the following **range 23 53**. The first port number in the range must be lower than the last number in the range. The *source-port number* and *destination-port-number* for the tcp-udp-operator is the number of the port.
ipv6-operator	Allows you to filter the packets further by using one of the following options: • **dscp** – The policy applies to packets that match the traffic class value in the traffic class field of the IPv6 packet header. This operator allows you to filter traffic based on TOS or IP precedence. You can specify a value from 0 – 63. • **fragments** – The policy applies to fragmented packets that contain a non-zero fragment offset. **NOTE:** This option is not applicable to filtering based on source or destination port, TCP flags, and ICMP flags. • **routing** – The policy applies only to IPv6 source-routed packets. **NOTE:** This option is not applicable to filtering based on source or destination port, TCP flags, and ICMP flags.

TABLE 18 Syntax descriptions (Continued)

IPv6 ACL arguments	Description
802.1p-priority-matching *number*	Enables the device to match only those packets that have the same 802.1p priorities as specified in the ACL. Enter 0 – 7. Use this option in conjunction with traffic policies to rate limit traffic for a specified 802.1p priority value. For details, refer to "Inspecting the 802.1p bit in the ACL for adaptive rate limiting" on page 146.
dscp-marking *number*	Use the **dscp-marking** *number* parameter to specify a new QoS value to the packet. **If a packet matches the filters in the ACL statement,** this parameter assigns the DSCP value that you specify to the packet. Enter 0 – 63.
802.1p-priority-marking *number*	Use the **802.1p-priority-marking** *number* parameter to specify a new QoS value to the packet (0-7). If a packet matches the filters in the ACL statement, it assigns the priority that you specify to the 802.1p priority and the internal priority.
internal-priority-marking *number*	Use the **internal-priority-marking** *number* parameter to specify a new QoS value to the packet (0-7). If a packet matches the filters in the ACL statement, it assigns the priority that you specify to the internal priority and the 802.1p priority. **NOTE:** Configuring **802.1p-priority-marking** alone or configuring both **802.1p-priority-marking** and **internal-priority-marking** has the same functionality. That is, it assigns the priority that you specify to the 802.1p priority and the internal priority.

ICMP message configurations

If you want to specify an ICMP message, you can enter one of the following ICMP message types:

- beyond-scope
- destination-unreachable
- echo-reply
- echo-request
- header
- hop-limit
- mld-query
- mld-reduction
- mld-report
- nd-na
- nd-ns
- next-header
- no-admin
- no-route
- packet-too-big
- parameter-option
- parameter-problem
- port-unreachable
- reassembly-timeout

- renum-command
- renum-result
- renum-seq-number
- router-advertisement
- router-renumbering
- router-solicitation
- time-exceeded
- unreachable

NOTE
If you do not specify a message type, the ACL applies to all types ICMP messages types.

Enabling IPv6 on an interface to which an ACL will be applied

Before an IPv6 ACL can be applied to an interface, it must first be created, and then IPv6 must be enabled on that interface.

To enable IPv6 on an interface, enter **ipv6 enable** at the Interface level of the CLI, or assign an IPv6 address to the interface, as described in *Brocade ICX 6650 Administration Guide*.

For example:

```
Brocade(config)# interface ethernet 1/1/1
Brocade(config-if-e10000-1/1/1)# ipv6 enable
```

These commands enable IPv6 on Ethernet interface 1/1/1 ready for an IPv6 ACL to be applied.

Syntax: ipv6 enable

When issued at the Interface Configuration level, this command enables IPv6 for a specific interface.

Applying an IPv6 ACL to an interface

As mentioned in "IPv6 ACL overview" on page 127, IPv6 ACLs are supported on the following devices:

- Gbps Ethernet ports
- 10 Gbps Ethernet ports
- Trunk groups
- Virtual routing interfaces

To apply an IPv6 ACL to an interface, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e100-1/3/1)# ipv6 traffic-filter access1 in
```

This example applies the IPv6 ACL “access1” to incoming IPv6 packets on Ethernet interface 1/3/1. As a result, Ethernet interface 1/3/1 denies all incoming packets from the site-local prefix 2001:db8:0:2::/64 and the global prefix 2001:db8:1::/48 and permits all other incoming packets.

Syntax for applying an IPv6 ACL

Syntax: .**ipv6 traffic-filter** *ipv6-ACL-name* **in**

For the *ipv6-ACL-name* parameter, specify the name of an IPv6 ACL created using the **ipv6 access-list** command.

The **in** keyword applies the specified IPv6 ACL to incoming IPv6 packets on the interface.

Applying an IPv6 ACL to a trunk group

When applying an IPv6 ACL to a trunk group, apply it to the primary port of the trunk, as described under “Applying an IPv6 ACL to an interface” on page 137. IPv6 ACLs cannot be applied to secondary ports. When an IPv6 ACL is applied to a primary port in a trunk, it filters the traffic on the secondary ports of the trunk as well as the traffic on the primary port.

Applying an IPv6 ACL to a virtual interface in a protocol-based or subnet-based VLAN

As with IPv4 ACLs, by default, when you apply an IPv6 ACL to a virtual interface in a protocol-based or subnet-based VLAN, the ACL takes effect on all protocol or subnet VLANs to which the untagged port belongs. To prevent the Brocade device from denying packets on other virtual interfaces that do not have an ACL applied, configure an ACL that permits packets in the IP subnet of the virtual interface in all protocol-based or subnet-based VLANs to which the untagged port belongs.

Adding a comment to an IPv6 ACL entry

You can optionally add a comment to describe entries in an IPv6 ACL. The comment appears in the output of **show** commands that display ACL information.

You can add a comment by entering the **remark** command immediately preceding an ACL entry, For example, to enter comments preceding an ACL entry, enter commands such as the following.

```
Brocade(config)# ipv6 access-list rtr
Brocade(config-ipv6-access-list rtr)# remark This entry permits ipv6 packets from
3002::2 to any destination
Brocade(config-ipv6-access-list rtr)# permit ipv6 host 2001:db8::2 any
Brocade(config-ipv6-access-list rtr)# remark This entry denies udp packets from
any source to any destination
Brocade(config-ipv6-access-list rtr)# deny udp any any
Brocade(config-ipv6-access-list rtr)# remark This entry denies IPv6 packets from
any source to any destination
Brocade(config-ipv6-access-list rtr)# deny ipv6 any any
Brocade(config-ipv6-access-list rtr)# write memory
```

Syntax: **remark** *comment-text*

The *comment-text* can be up to 256 characters in length.

The following shows the comment text for the ACL named "rtr" in a **show running-config** display.

```
Brocade# show running-config
ipv6 access-list rtr
 remark This entry permits ipv6 packets from 2001:db8::2 to any destination
 permit ipv6 host 2001:db8:1::2 any
 remark This entry denies udp packets from any source to any destination
 deny udp any any
 remark This entry denies IPv6 packets from any source to any destination
 deny ipv6 any any
```

Syntax: show running-config

Deleting a comment from an IPv6 ACL entry

To delete a comment from an IPv6 ACL entry, enter commands such as the following.

```
Brocade(config)# ipv6 access-list rtr
Brocade(config-ipv6-access-list rtr)# no remark This entry permits ipv6 packets
from 2001:db8:1::2 to any destination
```

Syntax: no remark *comment-text*

For *comment-text*, enter the text exactly as you did when you created the comment.

Support for ACL logging

Brocade devices support ACL logging of inbound packets that are sent to the CPU for processing (denied packets). ACL logging is not supported for any packets that are processed in hardware (permitted packets).

You may want the software to log entries in the syslog for packets that are denied by ACL filters. ACL logging is disabled by default; it must be explicitly enabled on a port. Refer to "ACL logging" on page 105.

Displaying IPv6 ACLs

To display the IPv6 ACLs configured on a device, enter the **show ipv6 access-list** command. Here is an example.

```
Brocade# show ipv6 access-list
ipv6 access-list v6-ACL1: 1 entries
 deny ipv6 any any
ipv6 access-list v6-ACL2: 1 entries
 permit ipv6 any any
ipv6 access-list v6-ACL3: 2 entries
 deny ipv6 2001:db8:10::/64 any
 permit ipv6 any any
ipv6 access-list v6-ACL4: 2 entries
 deny ipv6 2001:db8::/64 any
 permit ipv6 any any
ipv6 access-list rate-ACL: 1 entries
 permit ipv6 any any traffic-policy rate800M
ipv6 access-list v6-ACL5: 8 entries
 permit tcp 2001:db8::/64 any
 permit ipv6 2001:db8::/64 any
 permit ipv6 2001:db8:101::/64 any
 permit ipv6 2001:db8:10::/64 2001:db8:102::/64
 permit ipv6 host 2001:db8:10::102 host 2001:db8:101::102
 permit ipv6 host 2001:db8:10::101 host 2001:db8:101::101 dscp-matching 0
dscp-marking 63 dscp-cos-mapping
 permit ipv6 any any dscp-matching 63 dscp-cos-mapping
 permit ipv6 any any fragments
```

Syntax: show ipv6 access-list

To display a specific IPv6 ACL configured on a device, enter the **show ipv6 access-list** command followed by the ACL name. The following example shows the ACL named "rtr".

```
Brocade# show ipv6 access-list rtr
ipv6 access-list rtr: 3 entries
 remark This entry permits ipv6 packets from 2001:db8::2 to any destination
 permit ipv6 host 2001:db8:1::2 any
 remark This entry denies udp packets from any source to any destination
 deny udp any any
 remark This entry denies IPv6 packets from any source to any destination
 deny ipv6 any any
```

Syntax: show ipv6 access-list [*access-list-name*]

For the *access-list-name* parameter, specify the name of an IPv6 ACL created using the **ipv6 access-list** command.

Chapter

5

ACL-based Rate Limiting

Table 19 lists the ACL-based rate limiting features supported on Brocade ICX 6650. These features are supported in the Layer 2, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 19 Supported ACL-based rate limiting features

Feature	Brocade ICX 6650
Traffic policies	Yes
ACL-based fixed rate limiting	Yes
ACL-based adaptive rate limiting	Yes
802.1p priority bit inspection in the ACL for adaptive rate limiting	Yes
ACL statistics	Yes

ACL-based rate limiting overview

ACL-based rate limiting is a method for restricting inbound IP traffic that was permitted by extended IP ACLs, to administrator-configured rates. ACL-based rate limiting is available in the Layer 2 and Layer 3 images.

ACL-based rate limiting is defined using traffic policies. To configure ACL-based rate limiting, you create a traffic policy, reference the traffic policy in one or more ACL entries, and bind the ACL to an interface or port. The traffic policies become effective on ports to which the ACL is bound.

You can configure ACL-based rate limiting on the following interface types:

- Physical Ethernet interfaces
- Virtual interfaces
- Trunk ports
- Specific VLAN members on a port
- A subset of ports on a virtual interface

Types of ACL-based rate limiting

ACL-based rate limiting is of two types:

- Fixed rate limiting – Enforces a strict bandwidth limit. Traffic that exceeds the configured rate limit is either dropped or forwarded at the lowest priority level, depending on the action specified in the traffic policy. To configure fixed rate limiting, refer to "Configuring fixed rate limiting" on page 143.

- Adaptive rate limiting – Enforces a flexible bandwidth limit that allows for bursts above the limit. You can configure adaptive rate limiting to forward traffic, modify the IP precedence of and forward traffic, or drop traffic based on whether the traffic is within the limit or exceeds the limit. To configure adaptive rate limiting, refer to "Configuring adaptive rate limiting" on page 144.

Traffic policies overview

Traffic policies are rules that define rate limits on packets permitted by ACLs. As traffic policies apply rate limits on specific interfaces using ACLs, this method is also called ACL-based rate limiting. The process for applying a traffic policy to an interface involves:

1. Creating a traffic policy
2. Adding a reference to the traffic policy in an ACL entry
3. Binding the ACL associated with this ACL entry to an interface

Traffic policy structure

A traffic policy has the following structure:

- Traffic policy name – A string of up to eight alphanumeric characters that identifies individual traffic policy definitions.
- Traffic policy definition (TPD) – The command action associated with a traffic policy name. A TPD includes either or both of the following:
  - Rate limiting policy
  - ACL statistics

ACL statistics

Traffic policies also enable ACL statistics. ACL statistics, also called ACL counting, are automatically enabled when a traffic policy that defines a rate limit is enforced (activated). However, you can also create and enforce traffic policies that enable ACL statistics but do not enforce any rate limit.

On Brocade ICX 6650, ACL counting for fixed rate limiting is similar to the single-rate three-color marker (srTCM) mechanism described in RFC 2697. ACL counting for adaptive rate limiting is similar to the two-rate three-color marker (trTCM) mechanism described in RFC 2698.

In both types of rate limiting, ACL statistics can collect the following information:

- The total number of packets and bytes permitted by all ACLs to which the traffic policy is attached. This statistic is available in all traffic policies.
- The total number of packets at different conformance levels (either trTCM or srTCM, depending on the type of rate limiting applied) across all active ACLs to which the traffic policy is attached. This statistic is available in traffic policies that enable rate limiting.

NOTE
Refer to "Enabling and using ACL statistics" on page 148. To configure traffic policies for ACL counting, refer to "Enabling ACL statistics" on page 149.

Configuration notes for traffic policies

Consider the following points carefully before configuring traffic policies:

- Traffic policies apply to IP ACLs only.
- The maximum number of active TPDs (traffic policy definitions) supported by Brocade ICX 6650 is 896.
- You can reference the same traffic policy in more than one ACL entry within an ACL. For example, two or more ACL statements in ACL 101 can reference a TPD named TPD1.
- You can reference the same traffic policy in more than one ACL. For example, ACLs 101 and 102 could both reference a TPD named TPD1.
- Rate limits and ACL counting are applied at the traffic policy level, and are cumulative across ACLs and ACL entries on which they are applied. However, they are not cumulative across port regions. As Brocade ICX 6650 has a single port region, traffic policies defined on Brocade ICX 6650 are cumulative across the device.
- For all types of rate limiting on Brocade ICX 6650 (ACL-based; Port-based; and Broadcast, unknown Unicast, and Multicast rate limiting) the minimum value is 125 packets and can be increased in steps of 125 packets.
- To modify or delete an active traffic policy, you must first unbind the ACL that references the traffic policy.

Configuring fixed rate limiting

Fixed rate limiting enforces a strict bandwidth limit. The port forwards traffic that is within the limit. If the port receives more than the specified number of fragments in a one-second interval, the device either drops or forwards subsequent fragments in hardware, depending on the exceed action you specify.

NOTE
For related information on traffic policy features and limitations, see "Configuration notes for traffic policies" on page 143.

Follow these steps to implement the ACL-based fixed rate limiting.

1. Create a traffic policy. Enter a command such as the following:

```
Brocade(config)# traffic-policy TPD1 rate-limit fixed 125 exceed-action drop
```

2. Create an extended ACL entry (or modify an existing extended ACL entry) with a reference to the traffic policy. Enter a command such as the following.

```
Brocade(config)# access-list 101 permit ip host 10.10.12.2 any traffic-policy
TPD1
```

3. Bind the ACL to an interface. Enter commands such as the following.

```
Brocade(config)# interface ethernet 1/1/5
Brocade(config-if-e10000-1/1/5)# ip access-group 101 in
Brocade(config-if-e10000-1/1/5)# exit
```

The previous commands configure a fixed rate limiting policy that allows port e5 to receive a maximum traffic rate of 125 packets/second. If the port receives additional packets during a given one-second interval, the port drops the additional inbound packets received within that one-second interval.

Syntax: [**no**] **traffic-policy** *TPD-name* **rate-limit fixed** *cir-value* **exceed-action** *action* [**count**]

Syntax: **access-list** *num* **permit** | **deny.... traffic policy** *TPD-name*

Syntax: [**no**] **ip access-group** *num* **in**

NOTE
For brevity, the **access-list** command does not include all parameters.

ATTENTION
Brocade ICX 6650 allows you to add a reference to a non-existent TPD in an ACL statement and to bind that ACL to an interface. Brocade ICX 6650 does not issue a warning or an error message for non-existent TPDs.

Use the **no** form of the command to delete a traffic policy definition. Note that you cannot delete a traffic policy definition if it is currently in use on a port. To delete a traffic policy, first unbind the associated ACL.

The **traffic-policy** *TPD-name* parameter is the name of the traffic policy definition. This value can be eight or fewer alphanumeric characters.

The **rate-limit fixed** *cir-value* parameter enforces a strict bandwidth.The *cir-value* variable is the committed information rate in packets per second. This value can be from 125 through 15,000,000 packets per second.

The **exceed-action** *action* parameter specifies what happens to packets that exceed the configured committed information rate (CIR) value. Refer to "Handling packets that exceed the rate limit" on page 147.

The **count** parameter is optional and enables ACL counting. Refer to "Enabling and using ACL statistics" on page 148.

Configuring adaptive rate limiting

Adaptive rate limiting enforces a flexible bandwidth limit. The port forwards traffic that is within the limit. If the port receives more than the specified number of fragments in a one-second interval, the device either drops or forwards subsequent fragments in hardware, depending on the exceed action you specify.

NOTE
For information on related traffic policy features and limitations, see "Configuration notes for traffic policies" on page 143.

Table 20 lists the configurable parameters for ACL-based adaptive rate limiting.

TABLE 20 ACL based adaptive rate limiting parameters

Parameter	Definition
Committed Information Rate (CIR)	The guaranteed rate of inbound traffic (in packets per second) allowed on a port.
Committed Burst Size (CBS)	The number of packets per second allowed in a burst before some packets exceed the committed information rate. Larger bursts are more likely to exceed the rate limit. The CBS must be a value greater than zero (0). Brocade recommends that this value be equal to or greater than the size of the largest possible IP packet in a stream.
Peak Information Rate (PIR)	The maximum packets/second rate for inbound traffic on a port. The PIR must be equal to or greater than the CIR.
Peak Burst Size (PBS)	The number of packets per second allowed in a burst before all packets exceed the peak information rate. The PBS must be a value greater than zero (0). Brocade recommends that this value be equal to or greater than the size of the largest possible IP packet in the stream.

If a port receives more than the configured packet rate, the port either drops or forwards subsequent data in hardware, depending on the action you specify.

Marking Class of Service parameters in adaptive rate limiting

When you create a TPD, explicit marking of CoS parameters, such as traffic class and 802.1p priority, are not available on the device. For a TPD defining rate limiting, the device re-marks CoS parameters based on the DSCP value in the packet header and the determined conformance level of the rate limited traffic, as shown in Table 21.

TABLE 21 CoS parameters for packets that use rate limiting traffic policies

Packet conformance level	Packet DSCP value	Traffic class and 802.1p priority
0 (Green) or 1 (Yellow)	0 – 7	0 (lowest priority queue)
	8 – 15	1
	16 – 23	2
	24 – 31	3
	32 – 39	4
	40 – 47	5
	48 – 55	6
	56 – 63	7 (highest priority queue)
2 (Red)	N/A	0 (lowest priority queue)

Follow the steps given below to implement ACL-based adaptive rate limiting.

1. Create a traffic policy. Enter a command such as the following.

```
Brocade(config)# traffic-policy TPDAfour rate-limit adaptive cir 10000 cbs
1600 pir 20000 pbs 4000 exceed-action drop
```

2. Create a new extended ACL entry or modify an existing extended ACL entry that references the traffic policy. Enter a command such as the following.

```
Brocade(config)# access-list 104 permit ip host 10.10.12.2 any traffic-policy
TPDAfour
```

3. Bind the ACL to an interface. Enter commands such as the following.

```
Brocade(config)# interface ethernet 1/1/7
Brocade(config-if-e10000-1/1/7)# ip access-group 104 in
Brocade(config-if-e10000-1/1/7)# exit
```

The previous commands configure an adaptive rate limiting policy that enforces a guaranteed committed rate of 10000 packets/second on port e1/1/7 and allows bursts of up to 1600 packets. These commands also enforce a peak rate of 20000 packets/second and allow bursts of 4000 packets above the PIR limit. If the port receives additional bits during a given one-second interval, the port drops all packets on the port until the next one-second interval starts.

Syntax: [**no**] **traffic-policy** *TPD-name* **rate-limit adaptive cir** *cir-value* **cbs** *cbs-value* **pir** *pir-value* **pbs** *pbs-value* **exceed-action** *action* [**count**]

Syntax: **access-list** *num* **permit** | **deny.... traffic policy** *TPD-name*

Syntax: [**no**] **ip access-group** *num* **in**

NOTE
For brevity, the **access-list** command does not include all parameters.

ATTENTION
Brocade ICX 6650 allows you to add a reference to a non-existent TPD in an ACL statement and to bind that ACL to an interface. Brocade ICX 6650 does not issue a warning or an error message for non-existent TPDs.

Use the **no** form of the command to delete a traffic policy definition. Note that you cannot delete a traffic policy definition if it is currently in use on a port. To delete a traffic policy, first unbind the associated ACL.

The **traffic-policy** *TPD-name* parameter is the name of the traffic policy definition. This value can be eight or fewer alphanumeric characters.

The **rate-limit adaptive cir** *cir-value* option specifies that the policy will enforce a flexible bandwidth limit that allows for bursts above the limit.The *cir-value* parameter is the committed information rate in packets/second. Refer to Table 20.

The **cbs** *cbs-value* parameter is the committed burst size in packets. Refer to Table 20.

The **pir** *pir-value* parameter is the peak information rate in packets/second. Refer to Table 20.

The **pbs** *pbs-value* parameter is the peak burst size in packets. Refer to Table 20.

The **exceed-action** *action* parameter specifies the action taken on packets that exceed the configured values. Refer to "Handling packets that exceed the rate limit" on page 147.

The **count** parameter is optional and enables ACL statistics. Refer to "Enabling and using ACL statistics" on page 148.

Inspecting the 802.1p bit in the ACL for adaptive rate limiting

You can configure the Brocade device to rate limit traffic for a specified 802.1p priority value. To do so, complete the following configuration steps.

1. Create an adaptive rate limiting traffic policy. Enter command such as the following:

```
Brocade(config)# traffic-policy adap rate-limit adaptive cir 1000 cbs 1000 pir
2000 pbs 10000 exceed-action drop
```

2. Create an IPv4 extended ACL or IPv6 ACL that includes the traffic policy and 802.1p priority matching value. Enter a command such as the following:

```
Brocade(config)# access-list 136 permit ip any any 802.1p-priority matching 3
traffic-policy adap
```

3. Bind the ACL to an interface. Enter commands such as the following,.

```
Brocade(config)# interface ethernet 1/1/7
Brocade(config-if-e10000-1/1/7)# ip access-group 136 in
Brocade(config-if-e10000-1/1/7)# exit
```

Use the **show access-list accounting** command to view accounting statistics.

Handling packets that exceed the rate limit

For every traffic policy, you can specify what action to take on packets that exceed the configured rate limit. For both types of policies (fixed and adaptive rate limiting), you can specify one of the following actions:

- Drop packets that exceed the limit.
- Forward packets at the lowest priority level.

NOTE
For information on the how to configure a rate limit in fixed rate limiting, see"Configuring fixed rate limiting" on page 143. For information on the how to configure a rate limit in adaptive rate limiting, see"Configuring adaptive rate limiting" on page 144.

Dropping packets

The ultimate action that a device can take on a packet is to drop the packet. You can apply the drop action on packets that exceed the rate limit in both fixed rate limiting and adaptive rate limiting traffic policies. In fixed rate limiting policies, a packet is dropped only when the packet rate exceeds the CIR limit. Whereas, in adaptive rate limiting policies, a packet is dropped only when the packet rate exceeds PIR limit + PBS within one second.

The following example shows the drop action applied to a fixed rate limiting policy.

```
Brocade(config)# traffic-policy TPD1 rate-limit fixed 10000 exceed-action drop
```

The above command sets the fragment threshold at 10000 packets per second. If the port receives more than 10000 packets in a one-second interval, the device drops the excess fragments.

Syntax: [no] traffic-policy *TPD-name* **rate-limit fixed** *cir-value* **exceed-action drop**

The following example shows the drop action applied to an adaptive rate limiting policy.

```
Brocade(config)# traffic-policy TPDAfour rate-limit adaptive cir 10000 cbs 1600
pir 20000 pbs 4000 exceed-action drop
```

The above command creates an adaptive rate limiting policy that enforces a committed rate of 10000 packets per second with committed provisioning for burst sizes up to 1600 packets above the CIR limit. This command also enforces a peak rate of 20000 packets per second and allows bursts of 4000 packets above the PIR limit. If the port receives additional packets during a given one-second interval, the port drops all packets on the port until the next one-second interval starts.

Syntax: **[no] traffic-policy** *TPD-name* **rate-limit adaptive cir** *cir-value* **cbs** *cbs-value* **pir** *pir-value* **pbs** *pbs-value* **exceed-action drop**

Permitting packets at low priority

The alternative to dropping packets that exceed the rate limit, is to forward them at the lowest priority.

The following example shows a fixed rate limiting policy with the permit-at-the-lowest-priority action.

```
Brocade(config)# traffic-policy TPD1 rate-limit fixed 10000 exceed-action
permit-at-low-pri
```

The above command sets the fragment threshold at 10000 packets per second. If the port receives more than 10000 packets in any one-second interval, the device forwards the excess fragments at the lowest priority level.

Syntax: [**no**] **traffic-policy** *TPD-name* **rate-limit fixed** *cir-value* **exceed-action permit-at-low-pri**

The following example shows the permit-at-the-lowest-priority action applied to an adaptive rate limiting policy.

```
Brocade(config)# traffic-policy TPDAfour rate-limit adaptive cir 10000 cbs 1600
pir 20000 pbs 4000 exceed-action permit-at-low-pri
```

The above command creates an adaptive rate limiting policy that enforces a committed rate of 10000 packets per second with committed provisioning for burst sizes up to 1600 packets above the CIR limit. The above command also enforces a peak rate of 20000 packets per second and allows bursts of 4000 packets above the PIR limit. If the port receives additional packets during a given one-second interval, the port forwards excess packets at the lowest priority level until the next one-second interval starts.

Syntax: **[no] traffic-policy** *TPD-name* **rate-limit adaptive cir** *cir-value* **cbs** *cbs-value* **pir** *pir-value* **pbs** *pbs-value* **exceed-action permit-at-low-pri**

Enabling and using ACL statistics

ACL statistics help administrators discover how an individual traffic policy affects traffic. ACL statistics are automatically enabled when a traffic policy that specifies a rate limit is applied. However, you can also create (and apply) a traffic policy that enables ACL statistics without enforcing any rate limit.

"Enabling ACL statistics" on page 149 explains how to enable ACL statistics without implementing any rate limit. "Enabling ACL statistics with rate limiting traffic policies" on page 150 explains how to enable ACL statistics in a traffic policy that specifies a rate limit. "Viewing traffic policies" on page 152 explains how to view ACL statistics using **show** commands. "Clearing ACL and rate limit counters" on page 151 explains how to clear ACL statistic counters.

Enabling ACL statistics

The procedure for enabling ACL statistics is similar to the procedure for applying a rate limit: first create a traffic policy, then reference the traffic policy in an extended ACL entry, and finally bind the ACL to an interface. The ACL counting policy becomes effective on ports to which the ACLs are bound.

You also can enable ACL statistics when you create a traffic policy for rate limiting. Refer to "Enabling ACL statistics with rate limiting traffic policies" on page 150.

Follow these steps to enable ACL statistics without applying a rate limit.

1. Create a traffic policy. Enter a command such as the following.

```
Brocade(config)# traffic-policy TPD5 count
```

2. Create an extended ACL entry or modify an existing extended ACL entry that references the traffic policy definition. Enter a command such as the following.

```
Brocade(config)# access-list 101 permit ip host 10.10.12.2 any traffic-policy
TPD5
```

3. Bind the ACL to an interface. Enter commands such as the following.

```
Brocade(config)# interface ethernet 1/1/4
Brocade(config-if-e10000-1/1/4)# ip access-group 101 in
Brocade(config-if-e10000-1/1/4)# exit
```

The previous commands configure an ACL counting policy and apply it to port e1/1/4. Port e1/1/4 counts the number of packets and the number of bytes on the port that were permitted or denied by ACL filters.

Syntax: [**no**] **traffic-policy** *TPD-name* **count**

Syntax: **access-list** *num* **permit** | **deny**.... **traffic policy** *TPD-name*

Syntax: [**no**] **ip access-group** *num* **in**

NOTE
For brevity, some parameters were omitted from the **access-list** syntax.

ATTENTION
Brocade ICX 6650 allows you to add a reference to a non-existent TPD in an ACL statement and to bind that ACL to an interface. Brocade ICX 6650 does not issue a warning or an error message for non-existent TPDs.

Use the **no** form of the command to delete a traffic policy definition. Note that you cannot delete a traffic policy definition if it is currently in use on a port. To delete a traffic policy, first unbind the associated ACL.

The *TPD-name* variable is the name of the traffic policy definition. This value can be eight alphanumeric characters or less.

Enabling ACL statistics with rate limiting traffic policies

The configuration example in the section "Enabling ACL statistics" on page 149 shows how to enable ACL counting without applying rate limiting. You also can enable ACL counting while defining a rate limiting traffic policy.

To enable ACL counting while defining traffic policies for fixed rate limiting, enter the following commands at the global CONFIG level of the CLI.

```
Brocade(config)# traffic-policy TPD1 rate-limit fixed 1000 count
Brocade(config)# traffic-policy TPD2 rate-limit fixed 10000 exceed-action drop
count
```

Syntax: [**no**] **traffic-policy** *TPD-name* **rate-limit fixed** *cir-value* **count**

Syntax: [**no**] **traffic-policy** *TPD-name* **rate-limit fixed** *cir-value* **exceed-action** *action* **count**

To enable ACL counting while defining traffic policies for adaptive rate limiting, enter the following commands at the global CONFIG level of the CLI.

```
Brocade(config)# traffic-policy TPDA4 rate-limit adaptive cir 10000 cbs 1600 pir
20000 pbs 4000 count
Brocade(config)# traffic-policy TPDA5 rate-limit adaptive cir 10000 cbs 1600 pir
20000 pbs 4000 exceed-action permit-at-low-pri count
```

Syntax: **[no] traffic-policy** *TPD-name* **rate-limit adaptive cir** *cir-value* **cbs** *cbs-value* **pir** *pir-value* **pbs** *pbs-value* **count**

Syntax: **[no] traffic-policy** *TPD-name* **rate-limit adaptive cir** *cir-value* **cbs** *cbs-value* **pir** *pir-value* **pbs** *pbs-value* **exceed-action** *action* **count**

Viewing ACL and rate limit counters

When ACL counting is enabled on Brocade ICX 6650, you can use **show** commands to display the total packet count and byte count of the traffic filtered by ACL statements. The output of the **show** commands also displays the rate limiting traffic counters, which are automatically enabled for active rate limiting traffic policies.

Use either the **show access-list accounting traffic-policy** command or the **show statistics traffic-policy** command to display ACL and traffic policy counters. The output of both commands is identical.

The following example shows output from a **show access-list accounting** command.

```
Brocade#show access-list accounting traffic-policy tf125c
Traffic Policy tf125c:
Port Regions:
---------------
0 : 1/1/1-1/1/56, 1/3/1-1/3/8, 1/2/1-1/2/4

 General Counters:
 Port Region#            Byte Count          Packet Count
 ----------------  --------------------  --------------------
 7 (4/1 - 4/12)           85367040                776064
 All port regions         84367040                776064

 Rate Limiting Counters (in Packets):
```

```
Port Region#    Green Conformance   Yellow Conformance   Red Conformance
------------      -----------------    ------------------     ------------
7 (4/1 - 4/12)           551475                224589                0
All port regions         551475                224589                0
```

Syntax: show access-list accounting traffic-policy [*TPD-name*]

or

Syntax: show statistics traffic-policy [*TPD-name*]

The *TPD-name* variable is the name of the traffic policy definition for which you want to display ACL and traffic policy counters.

Table 22 explains the output of the **show access-list accounting traffic-policy** and **show statistics traffic-policy** commands.

TABLE 22 ACL and rate limit counting statistics

Parameter	Description
Traffic Policy	The name of the traffic policy.
General Counters	
Port Region #	The port region to which the active traffic policy applies.
Byte Count	The number of packets that were filtered (matched ACL clauses).
Packet Count	The number of packets that were filtered (matched ACL clauses).
Rate Limiting Counters	
Port Region#	The port region to which the active traffic policy applies.
Green Conformance	The number of packets that did not exceed the CIR packet rate.
Yellow Conformance	The number of packets that exceeded the CIR packet rate.
Red Conformance	The number of packets that exceeded the PIR packet rate.

Clearing ACL and rate limit counters

Brocade ICX 6650 keeps a running tally of the number of packets and the number of bytes per packet that are filtered by ACL statements and rate limiting traffic policies. You can clear these accumulated counters, essentially resetting them to zero. To do so, use either the **clear access-list accounting traffic-policy** command or the **clear statistics traffic-policy** command.

To clear the counters for ACL counting and rate limit counting, enter either of the following commands.

```
Brocade(config)# clear access-list accounting traffic-policy CountOne
Brocade(config)# clear statistics traffic-policy CountTwo
```

Syntax: clear access-list accounting traffic-policy *TPD-name*

or

Syntax: clear statistics traffic-policy *TPD-name*

The *TPD-name* is the name of the traffic policy definition for which you want to clear traffic policy counters.

Viewing traffic policies

To view traffic policies that are currently defined on Brocade ICX 6650, enter the **show traffic-policy** command. The following example shows the output of this command.Table 23 explains the output of the **show traffic-policy** command.

```
Brocade# show traffic-policy t_voip
Traffic Policy - t_voip:
Metering Enabled, Parameters:
          Mode:  Adaptive Rate-Limiting
          cir: 100 Pkts/s,     cbs: 2000 Pkts,     pir: 200 Pkts/s,     pbs: 4000
Pkts
Counting Not Enabled
```

Syntax: show traffic-policy [*TPD-name*]

To display all traffic policies, enter the **show traffic-policy** command without entering a TPD name.

TABLE 23 Traffic policy information

Parameter	Description
Traffic Policy	The name of the traffic policy.
Metering	Shows whether or not rate limiting was configured as part of the traffic policy: • **Enabled** – The traffic policy includes a rate limiting configuration. • **Disabled** – The traffic policy does not include a rate limiting configuration.
Mode	If rate limiting is enabled, this field shows the type of metering enabled on the port: • Fixed Rate-Limiting • Adaptive Rate-Limiting
cir	The committed information rate, in packets, for the adaptive rate limiting policy.
cbs	The committed burst size, in packets per second, for the adaptive rate-limiting policy.
pir	The peak information rate, in packets, for the adaptive rate limiting policy.
pbs	The peak burst size, in packets per second, for the adaptive rate limiting policy.
Counting	Shows whether or not ACL counting was configured as part of the traffic policy: • **Enabled** – Traffic policy includes an ACL counting configuration. • **Not Enabled** – Traffic policy does not include an ACL traffic counting configuration.

Chapter

802.1X Port Security

6

Table 24 lists 802.1X port security features that are supported on Brocade ICX 6650. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 24 Supported 802.1X port security features

Feature	Brocade ICX 6650
802.1X port security	Yes
Multiple host authentication	Yes
EAP pass-through support	Yes
802.1X accounting	Yes
802.1X dynamic assignment for ACL, MAC address filter, and VLAN	Yes
Automatic removal of Dynamic VLAN for 802.1X ports	Yes
RADIUS timeout action	Yes
802.1X and multi-device port authentication on the same port	Yes
802.1X and sFlow • 802.1X username export support for encrypted and non-encrypted EAP types	Yes

IETF RFC support

Brocade ICX 6650 supports the IEEE 802.1X standard for authenticating devices attached to LAN ports. Using 802.1X port security, you can configure a Brocade ICX 6650 device to grant access to a port based on information supplied by a client to an authentication server.

When a user logs on to a network that uses 802.1X port security, the Brocade device grants (or does not grant) access to network services after the user is authenticated by an authentication server. The user-based authentication in 802.1X port security provides an alternative to granting network access based on a user IP address, MAC address, or subnetwork.

The Brocade implementation of 802.1X port security supports the following RFCs:

- RFC 2284 PPP Extensible Authentication Protocol (EAP)
- RFC 2865 Remote Authentication Dial In User Service (RADIUS)
- RFC 2869 RADIUS Extensions

How 802.1X port security works

This section explains the basic concepts behind 802.1X port security, including device roles, how the devices communicate, and the procedure used for authenticating clients.

NOTE
802.1X Port Security cannot be configured on MAC Port Security-enabled ports.

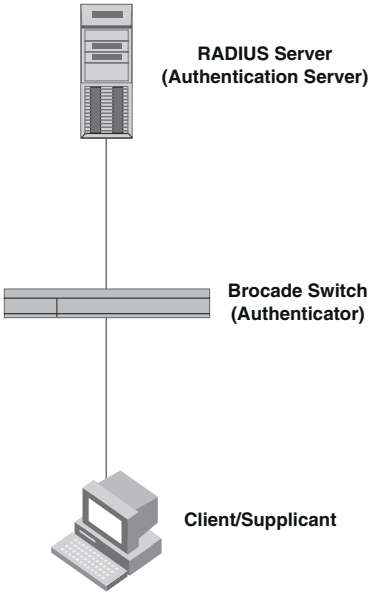
Device roles in an 802.1X configuration

The 802.1X standard defines the roles of ***Client/Supplicant***, ***Authenticator***, and ***Authentication Server*** in a network.

The Client (known as a ***Supplicant*** in the 802.1X standard) provides username/password information to the Authenticator. The Authenticator sends this information to the Authentication Server. Based on the Client's information, the Authentication Server determines whether the Client can use services provided by the Authenticator. The Authentication Server passes this information to the Authenticator, which then provides services to the Client, based on the authentication result.

Figure 1 illustrates these roles.

FIGURE 1 Authenticator, client/supplicant, and authentication server in an 802.1X configuration



Authenticator – The device that controls access to the network. In an 802.1X configuration, the Brocade device serves as the Authenticator. The Authenticator passes messages between the Client and the Authentication Server. Based on the identity information supplied by the Client, and the authentication information supplied by the Authentication Server, the Authenticator either grants or does not grant network access to the Client.

Client/Supplicant – The device that seeks to gain access to the network. Clients must be running software that supports the 802.1X standard (for example, the Windows XP operating system). Clients can either be directly connected to a port on the Authenticator, or can be connected by way of a hub.

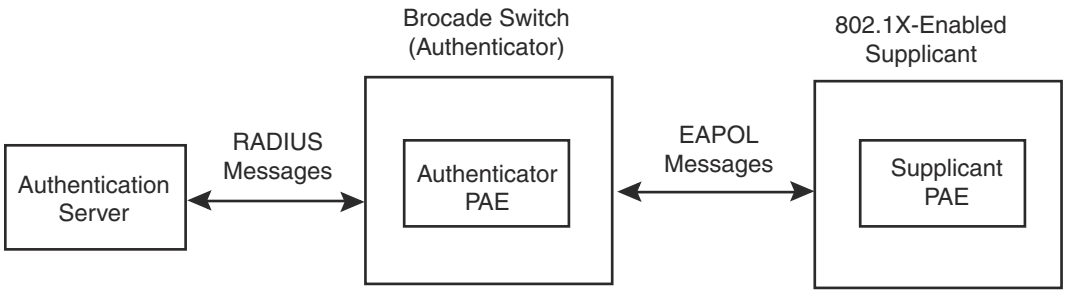
Authentication server – The device that validates the Client and specifies whether or not the Client may access services on the device. Brocade supports Authentication Servers running RADIUS.

Communication between the devices

For communication between the devices, 802.1X port security uses the ***Extensible Authentication Protocol*** (EAP), defined in RFC 2284. The 802.1X standard specifies a method for encapsulating EAP messages so that they can be carried over a LAN. This encapsulated form of EAP is known as EAP over LAN (***EAPOL***). The standard also specifies a means of transferring the EAPOL information between the Client/Supplicant, Authenticator, and Authentication Server.

EAPOL messages are passed between the ***Port Access Entity (PAE)*** on the Supplicant and the Authenticator. Figure 2 shows the relationship between the Authenticator PAE and the Supplicant PAE.

FIGURE 2 Authenticator PAE and supplicant PAE



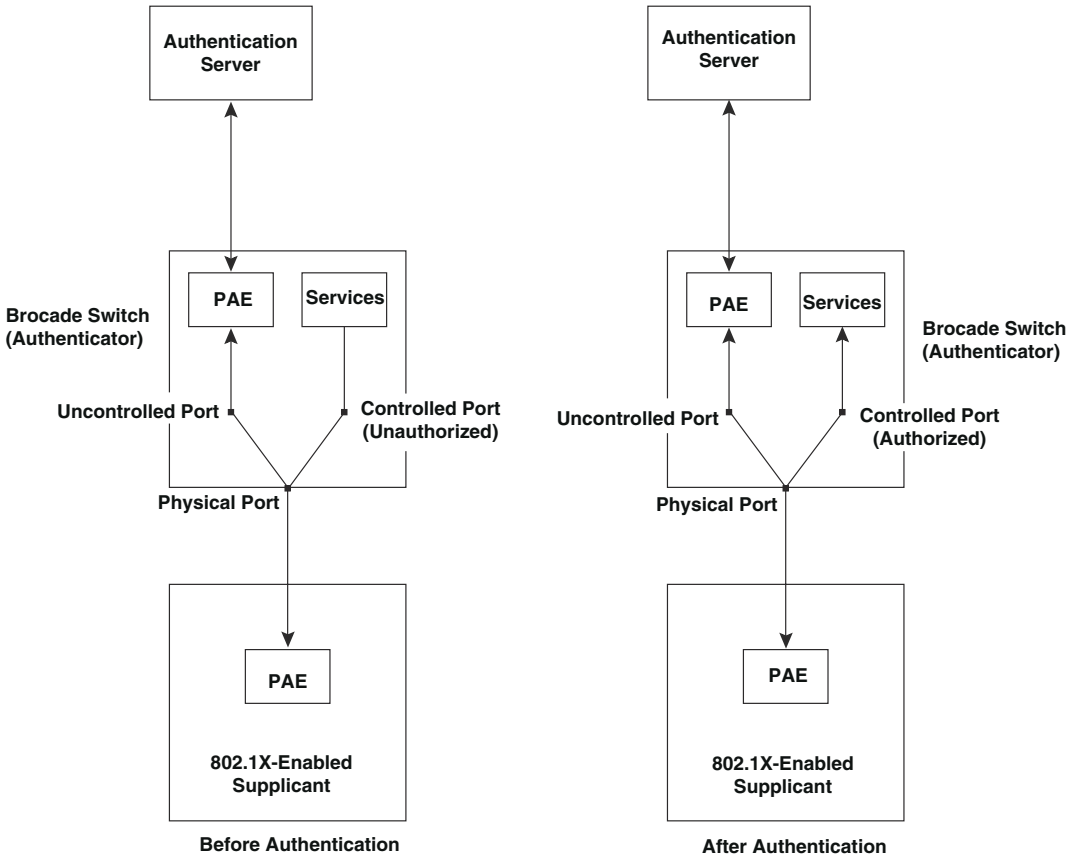
Authenticator PAE – The Authenticator PAE communicates with the Supplicant PAE, receiving identifying information from the Supplicant. Acting as a RADIUS client, the Authenticator PAE passes the Supplicant information to the Authentication Server, which decides whether the Supplicant can gain access to the port. If the Supplicant passes authentication, the Authenticator PAE grants it access to the port.

Supplicant PAE – The Supplicant PAE supplies information about the Client to the Authenticator PAE and responds to requests from the Authenticator PAE. The Supplicant PAE can also initiate the authentication procedure with the Authenticator PAE, as well as send log off messages.

Controlled and uncontrolled ports

A physical port on the device used with 802.1X port security has two virtual access points: a ***controlled*** port and an ***uncontrolled*** port. The controlled port provides full access to the network. The uncontrolled port provides access only for EAPOL traffic between the Client and the Authentication Server. When a Client is successfully authenticated, the controlled port is opened to the Client. Figure 3 illustrates this concept.

FIGURE 3 Controlled and uncontrolled ports before and after client authentication



Before a Client is authenticated, only the uncontrolled port on the Authenticator is open. The uncontrolled port allows only EAPOL frames to be exchanged between the Client and the Authentication Server. The controlled port is in the unauthorized state and allows no traffic to pass through.

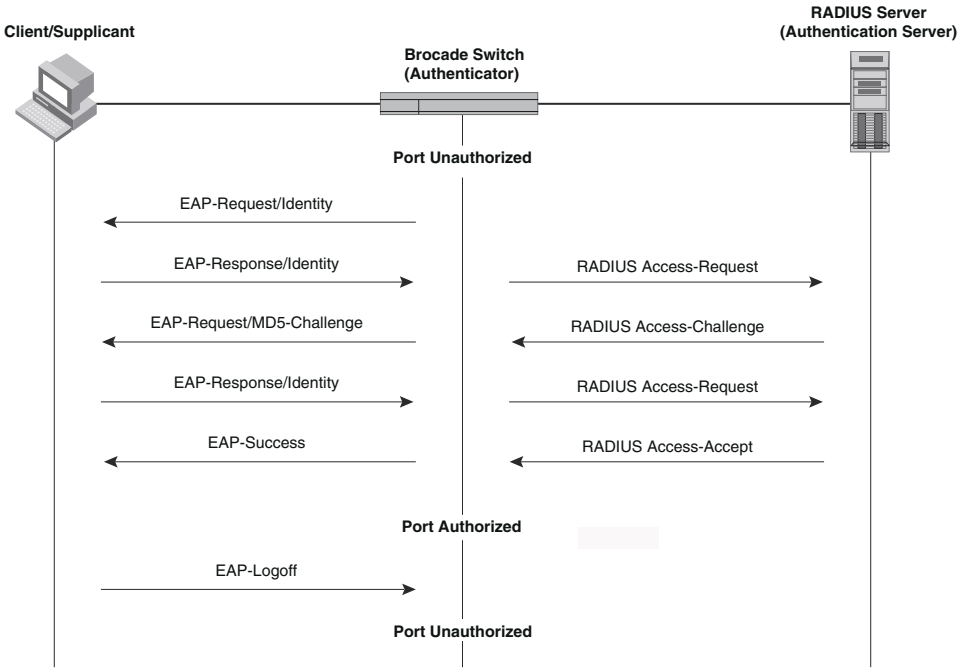
During authentication, EAPOL messages are exchanged between the Supplicant PAE and the Authenticator PAE, and RADIUS messages are exchanged between the Authenticator PAE and the Authentication Server.Refer to "Message exchange during authentication" on page 157 for an example of this process. If the Client is successfully authenticated, the controlled port becomes authorized, and traffic from the Client can flow through the port normally.

By default, all controlled ports on the Brocade device are placed in the authorized state, allowing all traffic. When authentication is activated on an 802.1X-enabled interface, the interface controlled port is placed initially in the unauthorized state. When a Client connected to the port is successfully authenticated, the controlled port is then placed in the authorized state until the Client logs off. Refer to "Enabling 802.1X port security" on page 174 for more information.

Message exchange during authentication

Figure 4 illustrates a sample exchange of messages between an 802.1X-enabled Client, a Brocade ICX 6650 switch acting as Authenticator, and a RADIUS server acting as an Authentication Server.

FIGURE 4 Message exchange between client/supplicant, authenticator, and authentication server



In this example, the Authenticator initiates communication with an 802.1X-enabled Client. When the Client responds, it is prompted for a username (255 characters maximum) and password. The Authenticator passes this information to the Authentication Server, which determines whether the Client can access services provided by the Authenticator. When the Client is successfully authenticated by the RADIUS server, the port is authorized. When the Client logs off, the port becomes unauthorized again.

The Brocade 802.1X implementation supports dynamic VLAN assignment. If one of the attributes in the Access-Accept message sent by the RADIUS server specifies a VLAN identifier, and this VLAN is available on the Brocade device, the client port is moved from its default VLAN to the specified VLAN. When the client disconnects from the network, the port is placed back in its default VLAN.Refer to "Dynamic VLAN assignment for 802.1X port configuration" on page 166 for more information.

If a Client does not support 802.1X, authentication cannot take place. The Brocade device sends EAP-Request/Identity frames to the Client, but the Client does not respond to them.

When a Client that supports 802.1X attempts to gain access through a non-802.1X-enabled port, it sends an EAP start frame to the Brocade device. When the device does not respond, the Client considers the port to be authorized, and starts sending normal traffic.

Brocade devices support Identity and MD5-challenge requests in EAP Request/Response messages as well as the following 802.1X authentication challenge types:

NOTE
Refer to "EAP pass-through support" on page 159.

- **EAP-TLS (RFC 2716)** – EAP Transport Level Security (TLS) provides strong security by requiring both client and authentication server to be identified and validated through the use of public key infrastructure (PKI) digital certificates. EAP-TLS establishes a tunnel between the client and the authentication server to protect messages from unauthorized users' eavesdropping activities. Since EAP-TLS requires PKI digital certificates on both the clients and the authentication servers, the roll out, maintenance, and scalability of this authentication method is much more complex than other methods. EAP-TLS is best for installations with existing PKI certificate infrastructures.
- **EAP-TTLS (Internet-Draft)** – The EAP Tunnelled Transport Level Security (TTLS) is an extension of EAP-TLS Like TLS, EAP-TTLS provides strong authentication; however it requires only the authentication server to be validated by the client through a certificate exchange between the server and the client. Clients are authenticated by the authentication server using user names and passwords.

  A TLS tunnel can be used to protect EAP messages and existing user credential services such as Active Directory, RADIUS, and LDAP. Backward compatibility for other authentication protocols such as PAP, CHAP, MS-CHAP, and MS-CHAP-V2 are also provided by EAP-TTLS. EAP-TTLS is not considered foolproof and can be fooled into sending identity credentials if TLS tunnels are not used. EAP-TTLS is suited for installations that require strong authentication without the use of mutual PKI digital certificates.
- **PEAP (Internet-Draft)** – Protected EAP Protocol (PEAP) is an Internet-Draft that is similar to EAP-TTLS. PEAP client authenticates directly with the backend authentication server. The authenticator acts as a pass-through device, which does not need to understand the specific EAP authentication protocols.

  Unlike EAP-TTLS, PEAP does not natively support user name and password to authenticate clients against an existing user database such as LDAP. PEAP secures the transmission between the client and authentication server with a TLS encrypted tunnel. PEAP also allows other EAP authentication protocols to be used. It relies on the mature TLS keying method for its key creation and exchange. PEAP is best suited for installations that require strong authentication without the use of mutual certificates.

Configuration for these challenge types is the same as for the EAP-MD5 challenge type.

NOTE
If the 802.1X Client will be sending a packet that is larger than 1500 bytes, you must enable **jumbo** at the Global config level of the CLI. If the supplicant or the RADIUS server does not support jumbo frames and jumbo is enabled on the switch, you can set the CPU IP MTU size. Refer to "Setting the IP MTU size", next.

Setting the IP MTU size

When jumbo frames are enabled on a Brocade ICX 6650 device and the certificate in use is larger than the standard packet size of 1500 bytes, 802.1X authentication will not work if the supplicant or the RADIUS server does not support jumbo frames. In this case, you can change the IP MTU setting so that the certificate will be fragmented before it is forwarded to the supplicant or server for processing. This feature is supported in the Layer 2 switch code only. It is not supported in the Layer 3 router code.

To enable this feature, enter the following command at the Global CONFIG level of the CLI.

```
Brocade(config)# ip mtu 1500
```

Syntax: [**no**] **ip mtu** *num*

The *num* parameter specifies the MTU. Ethernet II packets can hold IP packets from 576–1500 bytes long. If jumbo mode is enabled, Ethernet II packets can hold IP packets from 576–10,222 bytes long. Ethernet SNAP packets can hold IP packets from 576–1492 bytes long. If jumbo mode is enabled, SNAP packets can hold IP packets from 576 to 10,214 bytes long. The default MTU is 1500 for Ethernet II packets and 1492 for SNAP packets.

EAP pass-through support

EAP pass-through is supported on Brocade ICX 6650 devices that have 802.1X enabled. EAP pass-through support is fully compliant with RFC 3748, in which, by default, compliant pass-through authenticator implementations forward EAP challenge request packets *of any type*, including those listed in the previous section.

Configuration notes for setting the IP MTU size

If the 802.1X supplicant or authentication server will be sending packets that are greater than 1500 MTU, you should configure the device to accommodate a larger buffer size, in order to reduce problems during initial setup. Refer to *Brocade ICX 6650 Layer 3 Routing Configuration Guide*.

Support for RADIUS user-name attribute in access-accept messages

Brocade 802.1X-enabled ports support the RADIUS user-name (type 1) attribute in the Access-Accept message returned during 802.1X authentication.

This feature is useful when the client/supplicant does not provide its user-name in the EAP-response/identity frame, and the username is key to providing useful information. For example, when the user-name attribute is sent in the Access-Accept message, it is then available for display in sFlow sample messages sent to a collector, and in the output of some show dot1x CLI commands, such as show dot1x mac-sessions.

This same information is sent as the “user-name” attribute of RADIUS accounting messages, and is sent to the RADIUS accounting servers.

To enable this feature, add the following attribute on the RADIUS server.

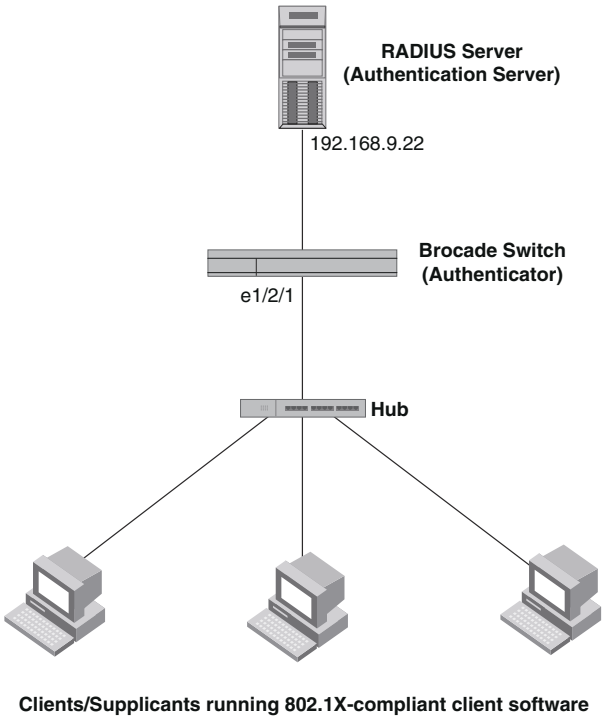
TABLE 25 RADIUS attributes

Attribute name	Type	Value
user-name	1	*name* (string)

Authenticating multiple hosts connected to the same port

Brocade devices support 802.1X authentication for ports with more than one host connected to them. Figure 5 illustrates a sample configuration where multiple hosts are connected to a single 802.1X port.

FIGURE 5 Multiple hosts connected to a single 802.1X-enabled port



If there are multiple hosts connected to a single 802.1X-enabled port, the Brocade device authenticates each of them individually. Each host authentication status is independent of the others, so that if one authenticated host disconnects from the network, it has no effect on the authentication status of any of the other authenticated hosts.

By default, traffic from hosts that cannot be authenticated by the RADIUS server is dropped in hardware. You can optionally configure the Brocade device to assign the port to a "restricted" VLAN if authentication of the Client is unsuccessful.

How 802.1X multiple-host authentication works

When multiple hosts are connected to a single 802.1X-enabled port on a Brocade device (as in Figure 5), 802.1X authentication is performed in the following way.

1. One of the 802.1X-enabled Clients attempts to log into a network in which a Brocade device serves as an Authenticator.
2. The Brocade device creates an internal session (called a ***dot1x-mac-session***) for the Client. A dot1x-mac-session serves to associate a Client MAC address and username with its authentication status.
3. The Brocade device performs 802.1X authentication for the Client. Messages are exchanged between the Brocade device and the Client, and between the device and the Authentication Server (RADIUS server). The result of this process is that the Client is either successfully authenticated or not authenticated, based on the username and password supplied by the client.
4. If the Client is successfully authenticated, the Client dot1x-mac-session is set to "access-is-allowed". This means that traffic from the Client can be forwarded normally.

5. If authentication for the Client is unsuccessful the first time, multiple attempts to authenticate the client will be made as determined by the **attempts** variable in the **auth-fail-max-attempts** command.
   - Refer to "Specifying the number of authentication attempts the device makes before dropping packets" on page 180 for information on how to do this.
6. If authentication for the Client is unsuccessful more than the number of times specified by the attempts variable in the **auth-fail-max-attempts** command, an **authentication-failure** action is taken. The authentication-failure action can be either to drop traffic from the Client, or to place the port in a "restricted" VLAN:
   - If the authentication-failure action is to drop traffic from the Client, then the Client dot1x-mac-session is set to "access-denied", causing traffic from the Client to be dropped in hardware.
   - If the authentication-failure action is to place the port in a "restricted" VLAN, If the Client dot1x-mac-session is set to "access-restricted" then the port is moved to the specified restricted VLAN, and traffic from the Client is forwarded normally.
7. When the Client disconnects from the network, the Brocade device deletes the Client dot1x-mac-session. This does not affect the dot1x-mac-session or authentication status (if any) of the other hosts connected on the port.

Configuration notes for 802.1x multiple-host authentication

- The Client dot1x-mac-session establishes a relationship between the username and MAC address used for authentication. If a user attempts to gain access from different Clients (with different MAC addresses), he or she would need to be authenticated from each Client.
- If a Client has been denied access to the network (that is, the Client dot1x-mac-session is set to "access-denied"), then you can cause the Client to be re-authenticated by manually disconnecting the Client from the network, or by using the **clear dot1x mac-session** command. Refer to "Clearing a dot1x-mac-session for a MAC address" on page 181 for information on this command.
- When a Client has been denied access to the network, its dot1x-mac-session is aged out if no traffic is received from the Client MAC address over a fixed hardware aging period (70 seconds), plus a configurable software aging period. You can optionally change the software aging period for dot1x-mac-sessions or disable aging altogether. After the denied Client dot1x-mac-session is aged out, traffic from that Client is no longer blocked, and the Client can be re-authenticated.

  In addition, you can configure disable aging for the dot1x-mac-session of Clients that have been granted either full access to the network, or have been placed in a restricted VLAN. After a Client dot1x-mac-session ages out, the Client must be re-authenticated.Refer to "Disabling aging for dot1x-mac-sessions" on page 180 for more information.
- Dynamic IP ACL and MAC address filter assignment is supported in an 802.1X multiple-host configuration. Refer to "Dynamically applying IP ACLs and MAC address filters to 802.1X ports" on page 170.

- 802.1X multiple-host authentication has the following additions:
  - Configurable hardware aging period for denied client dot1x-mac-sessions. Refer to "Configurable hardware aging period for denied client dot1x-mac-sessions" on page 162.
  - Dynamic ACL and MAC address filter assignment in 802.1X multiple-host configurations. Refer to "Dynamically applying IP ACLs and MAC address filters to 802.1X ports" on page 170.
  - Dynamic multiple VLAN assignment for 802.1X ports. Refer "Dynamic multiple VLAN assignment for 802.1X ports" on page 168.
  - Configure a restriction to forward authenticated and unauthenticated tagged and untagged clients to a restricted VLAN.
  - Configure an override to send failed dot1x and non-dot1x clients to a restricted VLAN.
  - Configure VLAN assignments for clients attempting to gain access through dual-mode ports.
  - Enhancements to some **show** commands.
  - Differences in command syntax for saving dynamic VLAN assignments to the startup-config file.

Configurable hardware aging period for denied client dot1x-mac-sessions

When one of the 802.1X-enabled Clients in a multiple-host configuration attempts to log into a network in which a Brocade device serves as an Authenticator, the device creates a dot1x-mac-session for the Client.

When a Client has been denied access to the network, its dot1x-mac-session is aged out if no traffic is received from the Client MAC address over a period of time. After a denied Client dot1x-mac-session ages out, the Client can be re-authenticated. Aging of a denied Client's dot1x-mac-session occurs in two phases, known as hardware aging and software aging.

The hardware aging period for a denied Client's dot1x-mac-session is not fixed at 70 seconds. The hardware aging period for a denied Client's dot1x-mac-session is equal to the length of time specified with the dot1x **timeout quiet-period** command. By default, the hardware aging time is 60 seconds. Once the hardware aging period ends, the software aging period begins. When the software aging period ends, the denied Client's dot1x-mac-session ages out, and the Client can be authenticated again.

802.1X port security and sFlow

sFlow is a standards-based protocol that allows network traffic to be sampled at a user-defined rate for the purpose of monitoring traffic flow patterns and identifying packet transfer rates on user-specified interfaces.

When you enable sFlow forwarding on an 802.1X-enabled interface, the samples taken from the interface include the user name string at the inbound or outbound port, or both, if that information is available.

For more information on sFlow, refer to the *Brocade ICX 6650 Administration Guide*.

802.1X accounting

When 802.1X port security is enabled on the Brocade device, you can enable ***802.1X accounting***. This feature enables the Brocade device to log information on the RADIUS server about authenticated 802.1X clients. The information logged on the RADIUS server includes the 802.1X client session ID, MAC address, and authenticating physical port number.

802.1X accounting works as follows.

1. A RADIUS server successfully authenticates an 802.1X client.
2. If 802.1X accounting is enabled, the Brocade device sends an 802.1X Accounting Start packet to the RADIUS server, indicating the start of a new session.
3. The RADIUS server acknowledges the Accounting Start packet.
4. The RADIUS server records information about the client.
5. When the session is concluded, the Brocade device sends an Accounting Stop packet to the RADIUS server, indicating the end of the session.
6. The RADIUS server acknowledges the Accounting Stop packet.

To enable 802.1X accounting, refer to "802.1X accounting configuration" on page 182.

802.1X port security configuration

Configuring 802.1X port security on a Brocade device consists of the following tasks.

1. Configure the device interaction with the Authentication Server:
   - "Configuring an authentication method list for 802.1X" on page 164
   - "Setting RADIUS parameters" on page 164
   - "Dynamic VLAN assignment for 802.1X port configuration" on page 166 (optional)
   - "Dynamically applying IP ACLs and MAC address filters to 802.1X ports" on page 170
2. Configure the device role as the Authenticator:
   - "Enabling 802.1X port security" on page 174
   - "Initializing 802.1X on a port" on page 178 (optional)
3. Configure the device interaction with Clients:
   - "Configuring periodic re-authentication" on page 175 (optional)
   - "Re-authenticating a port manually" on page 176 (optional)
   - "Setting the quiet period" on page 176 (optional)
   - "Setting the wait interval for EAP frame retransmissions" on page 176 (optional)
   - "Setting the maximum number of EAP frame retransmissions" on page 177 (optional)
   - "Specifying a timeout for retransmission of messages to the authentication server" on page 178 (optional)
   - "Allowing access to multiple hosts" on page 179 (optional)
   - "MAC address filters for EAP frames" on page 182 (optional)

Configuring an authentication method list for 802.1X

To use 802.1X port security, you must specify an authentication method to be used to authenticate Clients. Brocade supports RADIUS authentication with 802.1X port security. To use RADIUS authentication with 802.1X port security, you create an authentication method list for 802.1X and specify RADIUS as an authentication method, then configure communication between the Brocade device and RADIUS server.

Example

```
Brocade(config)# aaa authentication dot1x default radius
```

Syntax: [**no**] **aaa authentication dot1x default** *method-list*

For the *method-list*, enter at least one of the following authentication methods

radius – Use the list of all RADIUS servers that support 802.1X for authentication.

none – Use no authentication. The Client is automatically authenticated by other means, without the device using information supplied by the Client.

NOTE
If you specify both **radius** and **none**, make sure **radius** comes before **none** in the method list.

Setting RADIUS parameters

To use a RADIUS server to authenticate access to a Brocade device, you must identify the server to the Brocade device.

Example

```
Brocade(config)# radius-server host 10.157.22.99 auth-port 1812 acct-port 1813
default key mirabeau dot1x
```

Syntax: **radius-server host** *ip-addr* | *ipv6-addr* | *server-name* [**auth-port** *num* | **acct-port** *num* | **default**] [**key 0** | **1** *string*] [**dot1x**]

The **host** *ip-addr* | *ipv6-addr* | *server-name* parameter is either an IP address or an ASCII text string.

The **dot1x** parameter indicates that this RADIUS server supports the 802.1X standard. A RADIUS server that supports the 802.1X standard can also be used to authenticate non-802.1X authentication requests.

NOTE
To implement 802.1X port security, at least one of the RADIUS servers identified to the Brocade device must support the 802.1X standard.

Supported RADIUS attributes

Many IEEE 802.1X Authenticators will function as RADIUS clients. Some of the RADIUS attributes may be received as part of IEEE 802.1X authentication. Brocade devices support the following RADIUS attributes for IEEE 802.1X authentication:

- Username (1) – RFC 2865

- NAS-IP-Address (4) – RFC 2865
- NAS-Port (5) – RFC 2865
- Service-Type (6) – RFC 2865
- FilterId (11) – RFC 2865
- Framed-MTU (12) – RFC 2865
- State (24) – RFC 2865
- Vendor-Specific (26) – RFC 2865
- Session-Timeout (27) – RFC 2865
- Termination-Action (29) – RFC 2865
- Calling-Station-ID (31) – RFC 2865
- NAS-Port-Type (61) – RFC 2865
- Tunnel-Type (64) – RFC 2868
- Tunnel-Medium-Type (65) – RFC 2868
- EAP Message (79) – RFC 2579
- Message-Authenticator (80) RFC 3579
- Tunnel-Private-Group-Id (81) – RFC 2868
- NAS-Port-id (87) – RFC 2869

Specifying the RADIUS timeout action

A RADIUS timeout occurs when the Brocade device does not receive a response from a RADIUS server within a specified time limit and after a certain number of retries. The time limit and number of retries can be manually configured using the CLI commands **radius-server timeout** and **radius-server retransmit**, respectively. If the parameters are not manually configured, the Brocade device applies the default value of three seconds time limit with a maximum of three retries.

You can better control port behavior when a RADIUS timeout occurs. That is, you can configure a port on the Brocade device to automatically pass or fail users being authenticated. A ***pass*** essentially bypasses the authentication process and permits user access to the network. A ***fail*** bypasses the authentication process and blocks user access to the network, unless restrict-vlan is configured, in which case, the user is placed into a VLAN with restricted or limited access. By default, the Brocade device will reset the authentication process and retry to authenticate the user.

Specify the RADIUS timeout action at the Interface level of the CLI.

Permit user access to the network after a RADIUS timeout

To set the RADIUS timeout behavior to bypass 802.1X authentication and *permit* user access to the network, enter commands such as the following

```
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# dot1x auth-timeout-action success
```

Syntax: [no] **dot1x auth-timeout-action success**

Once the *success* timeout action is enabled, use the **no** form of the command to reset the RADIUS timeout behavior to *retry*.

Re-authenticate a user

To configure RADIUS timeout behavior to bypass multi-device port authentication and *permit* user access to the network, enter commands similar to the following

```
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# dot1x re-auth-timeout-success 60
```

Syntax: [**no**] **dot1x re-auth-timeout- success** *seconds*

The *seconds* parameter specifies the number of seconds the device will wait to re-authenticate a user after a timeout. The minimum value is 10 seconds. The maximum value is 2^{16}-1 (maximum unsigned 16-bit value).

Deny user access to the network after a RADIUS timeout

To set the RADIUS timeout behavior to bypass 802.1X authentication and *block* user access to the network, enter commands such as the following

```
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# dot1x auth-timeout-action failure
```

Syntax: [**no**] **dot1x auth-timeout-action failure**

Once the failure timeout action is enabled, use the **no** form of the command to reset the RADIUS timeout behavior to ***retry***.

NOTE
If **restrict-vlan** is configured along with **auth-timeout-action** failure, the user will be placed into a VLAN with restricted or limited access.Refer to "Allow user access to a restricted VLAN after a RADIUS timeout" on page 166.

Allow user access to a restricted VLAN after a RADIUS timeout

To set the RADIUS timeout behavior to bypass 802.1X authentication and place the user in a VLAN with restricted or limited access, enter commands such as the following

```
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# dot1x auth-timeout-action failure
```

Syntax: [**no**] **dot1x auth-timeout-action failure**

NOTE
The commands **auth-fail-action restrict-vlan** and **auth-fail-vlanid** are supported in the global dot1x mode and are not supported at the port-level. The failure action of **dot1x auth-timeout-action failure** will follow the **auth-fail-action** defined at the global dot1x level.

Dynamic VLAN assignment for 802.1X port configuration

When a client successfully completes the EAP authentication process, the Authentication Server (the RADIUS server) sends the Authenticator (the Brocade device) a RADIUS Access-Accept message that grants the client access to the network. The RADIUS Access-Accept message contains attributes set for the user in the user's access profile on the RADIUS server.

If one of the attributes in the Access-Accept message specifies a VLAN identifier, and if this VLAN is available on the Brocade device, the client port is moved from its default VLAN to this specified VLAN.

NOTE
This feature is supported on port-based VLANs only. This feature cannot be used to place an 802.1X-enabled port into a Layer 3 protocol VLAN.

Automatic removal of dynamic VLAN assignments for 802.1X ports

For increased security, this feature removes any association between a port and a dynamically-assigned VLAN when all 802.1x sessions for that VLAN have expired on the port.

NOTE
When a **show run** command is issued during a session, the dynamically-assigned VLAN is not displayed.

Enable 802.1X VLAN ID support by adding the following attributes to a user profile on the RADIUS server.

TABLE 26 802.1X VLAN ID attributes

Attribute name	Type	Value
Tunnel-Type	064	13 (decimal) – VLAN
Tunnel-Medium-Type	065	6 (decimal) – 802
Tunnel-Private-Group-ID	081	*vlan-name* (string) – either the name or the number of a VLAN configured on the Brocade device.

The device reads the attributes as follows:

- If the Tunnel-Type or the Tunnel-Medium-Type attributes in the Access-Accept message do not have the values specified above, the Brocade device ignores the three Attribute-Value pairs. The client becomes authorized, but the client port is not dynamically placed in a VLAN.
- If the Tunnel-Type or the Tunnel-Medium-Type attributes in the Access-Accept message do have the values specified above, but there is no value specified for the Tunnel-Private-Group-ID attribute, the client will not become authorized.
- When the Brocade device receives the value specified for the Tunnel-Private-Group-ID attribute, it checks whether the *vlan-name* string matches the name of a VLAN configured on the device. If there is a VLAN on the device whose name matches the *vlan-name* string, then the client port is placed in the VLAN whose ID corresponds to the VLAN name.
- If the *vlan-name* string does not match the name of a VLAN, the Brocade device checks whether the string, when converted to a number, matches the ID of a VLAN configured on the device. If it does, then the client port is placed in the VLAN with that ID.
- If the *vlan-name* string does not match either the name or the ID of a VLAN configured on the device, then the client will not become authorized.

The **show interface** command displays the VLAN to which an 802.1X-enabled port has been dynamically assigned, as well as the port from which it was moved (that is, the port default VLAN).Refer to "Displaying dynamically assigned VLAN information" on page 188 for sample output indicating the port dynamically assigned VLAN.

Dynamic multiple VLAN assignment for 802.1X ports

When you add attributes to a user profile on the RADIUS server, the *vlan-name* value for the Tunnel-Private-Group-ID attribute can specify the name or number of one or more VLANs configured on the Brocade device.

For example, to specify one VLAN, configure the following for the *vlan-name* value in the Tunnel-Private-Group-ID attribute on the RADIUS server.

"10" or "marketing"

In this example, the port on which the Client is authenticated is assigned to VLAN 10 or the VLAN named "marketing". The VLAN to which the port is assigned must have previously been configured on the Brocade device.

Specifying an untagged VLAN

To specify an untagged VLAN, use the following.

"U:10" or "U:marketing"

When the RADIUS server specifies an untagged VLAN ID, the port default VLAN ID (or ***PVID***) is changed from the system DEFAULT-VLAN (VLAN 1) to the specified VLAN ID. The port transmits only untagged traffic on its PVID. In this example, the port PVID is changed from VLAN 1 (the DEFAULT-VLAN) to VLAN 10 or the VLAN named “marketing”.

The PVID for a port can be changed only once through RADIUS authentication. For example, if RADIUS authentication for a Client causes a port PVID to be changed from 1 to 10, and then RADIUS authentication for another Client on the same port specifies that the port PVID be moved to 20, then the second PVID assignment from the RADIUS server is ignored.

If the link goes down, or the dot1x-mac-session for the Client that caused the initial PVID assignment ages out, then the port reverts back to its original (non-RADIUS-specified) PVID, and subsequent RADIUS authentication can change the PVID assignment for the port.

If a port PVID is assigned through the multi-device port authentication feature, and 802.1X authentication subsequently specifies a different PVID, then the PVID specified through 802.1X authentication overrides the PVID specified through multi-device port authentication.

Specifying a tagged VLAN

To specify a tagged VLAN, use the following.

"T:12;T:20" or "T:12;T:marketing"

In this example, the port is added to VLANs 12 and 20 or VLANs 12 and the VLAN named "marketing". When a tagged packet is authenticated, and a list of VLANs is specified on the RADIUS server for the MAC address, then the packet tag must match one of the VLANs in the list in order for the Client to be successfully authenticated. If authentication is successful, then the port is added to all of the VLANs specified in the list.

Unlike with a RADIUS-specified untagged VLAN, if the dot1x-mac-session for the Client ages out, the port membership in RADIUS-specified tagged VLANs is not changed. In addition, if multi-device port authentication specifies a different list of tagged VLANs, then the port is added to the specified list of VLANs. Membership in the VLANs specified through 802.1X authentication is not changed.

Specifying an untagged VLAN and multiple tagged VLANs

To specify an untagged VLAN and multiple tagged VLANs, use the following.

"U:10;T:12;T:marketing"

When the RADIUS server returns a value specifying both untagged and tagged VLAN IDs, the port becomes a dual-mode port, accepting and transmitting both tagged traffic and untagged traffic at the same time. A dual-mode port transmits only untagged traffic on its default VLAN (PVID) and only tagged traffic on all other VLANs.

In this example, the port VLAN configuration is changed so that it transmits untagged traffic on VLAN 10, and transmits tagged traffic on VLAN 12 and the VLAN named "marketing".

For a configuration example, refer to "802.1X authentication with dynamic VLAN assignment" on page 198.

Saving dynamic VLAN assignments to the running-config file

You can configure the Brocade device to save the RADIUS-specified VLAN assignments to the device's running-config file. Enter commands such as the following.

```
Brocade(config)# dot1x-enable
Brocade(config-dot1x)# save-dynamicvlan-to-config
```

Syntax: save-dynamicvlan-to-config

By default, the dynamic VLAN assignments are not saved to the running-config file. Entering the **show running-config** command does not display dynamic VLAN assignments, although they can be displayed with the s**how vlan** and **show authenticated-mac-address detail** commands.

NOTE
When this feature is enabled, issuing the command **write mem** will save any dynamic VLAN assignments to the startup configuration file.

Considerations for dynamic VLAN assignment in an 802.1X multiple-host configuration

The following considerations apply when a Client in a 802.1X multiple-host configuration is successfully authenticated, and the RADIUS Access-Accept message specifies a VLAN for the port:

- If the port is not already a member of a RADIUS-specified VLAN, and the RADIUS Access-Accept message specifies the name or ID of a valid VLAN on the Brocade device, then the port is placed in that VLAN.
- If the port is already a member of a RADIUS-specified VLAN, and the RADIUS Access-Accept message specifies the name or ID of a different VLAN, then it is considered an authentication failure. The port VLAN membership is not changed.
- If the port is already a member of a RADIUS-specified VLAN, and the RADIUS Access-Accept message specifies the name or ID of that same VLAN, then traffic from the Client is forwarded normally.
- If the RADIUS Access-Accept message specifies the name or ID of a VLAN that does not exist on the Brocade device, then it is considered an authentication failure.
- If the port is a tagged or dual-mode port, and the RADIUS Access-Accept message specifies the name or ID of a valid VLAN on the Brocade device, then the port is placed in that VLAN. If the port is already a member of the RADIUS-specified VLAN, no further action is taken.
- If the RADIUS Access-Accept message does not contain any VLAN information, the Client dot1x-mac-session is set to "access-is-allowed". If the port is already in a RADIUS-specified VLAN, it remains in that VLAN.

Dynamically applying IP ACLs and MAC address filters to 802.1X ports

The Brocade 802.1X implementation supports dynamically applying an IP ACL or MAC address filter to a port, based on information received from an Authentication Server.

When a client/supplicant successfully completes the EAP authentication process, the Authentication Server (the RADIUS server) sends the Authenticator (the Brocade device) a RADIUS Access-Accept message that grants the client access to the network. The RADIUS Access-Accept message contains attributes set for the user in the user's access profile on the RADIUS server.

If the Access-Accept message contains Filter-ID (type 11) or Vendor-Specific (type 26), or both attributes, the Brocade device can use information in these attributes to apply an IP ACL or MAC address filter to the authenticated port. This IP ACL or MAC address filter applies to the port for as long as the client is connected to the network. When the client disconnects from the network, the IP ACL or MAC address filter is no longer applied to the port. If an IP ACL or MAC address filter had been applied to the port prior to 802.1X authentication, it is then re-applied to the port.

The Brocade device uses information in the Filter ID and Vendor-Specific attributes as follows:

- The Filter-ID attribute can specify the number of an existing IP ACL or MAC address filter configured on the Brocade device. In this case, the IP ACL or MAC address filter with the specified number is applied to the port.
- The Vendor-Specific attribute can specify actual syntax for a Brocade IP ACL or MAC address filter, which is then applied to the authenticated port. Configuring a Vendor-Specific attribute in this way allows you to create IP ACLs and MAC address filters that apply to individual users; that is, *per-user* IP ACLs or MAC address filters.

Configuration considerations for applying IP ACLs and MAC address filters to 802.1x ports

The following restrictions apply to dynamic IP ACLs or MAC address filters:

- Inbound dynamic IP ACLs are supported. Outbound dynamic ACLs are not supported.
- Inbound Vendor-Specific attributes are supported. Outbound Vendor-Specific attributes are not supported.
- A maximum of one IP ACL can be configured in the inbound direction on an interface.
- 802.1X with dynamic MAC filter will work for one client at a time on a port. If a second client tries to authenticate with 802.1X and dynamic MAC filter, the second client will be rejected.
- MAC address filters cannot be configured in the outbound direction on an interface.
- Concurrent operation of MAC address filters and IP ACLs is not supported.
- A dynamic IP ACL will take precedence over an IP ACL that is bound to a port (port ACL). When a client authenticates with a dynamic IP ACL, the port ACL will not be applied. Also, future clients on the same port will authenticate with a dynamic IP ACL or no IP ACL. If no clients on the port use dynamic ACL, then the port ACL will be applied to all traffic.

Disabling and enabling strict security mode for dynamic filter assignment

By default, 802.1X dynamic filter assignment operates in ***strict security mode***. When strict security mode is enabled, 802.1X authentication for a port fails if the Filter-ID attribute contains invalid information, or if insufficient system resources are available to implement the per-user IP ACLs or MAC address filters specified in the Vendor-Specific attribute.

When strict security mode is enabled:

- If the Filter-ID attribute in the Access-Accept message contains a value that does not refer to an existing filter (that is, a MAC address filter or IP ACL configured on the device), then the port will not be authenticated, regardless of any other information in the message (for example, if the Tunnel-Private-Group-ID attribute specifies a VLAN on which to assign the port).
- If the Vendor-Specific attribute specifies the syntax for a filter, but there are insufficient system resources to implement the filter, then the port will not be authenticated.
- If the device does not have the system resources available to dynamically apply a filter to a port, then the port will not be authenticated.

NOTE
If the Access-Accept message contains values for both the Filter-ID and Vendor-Specific attributes, then the value in the Vendor-Specific attribute (the per-user filter) takes precedence.

Also, if authentication for a port fails because the Filter-ID attribute referred to a non-existent filter, or there were insufficient system resources to implement the filter, then a syslog message is generated.

Disabled strict security mode

When strict security mode is disabled:

- If the Filter-ID attribute in the Access-Accept message contains a value that does not refer to an existing filter (that is, a MAC address filter or IP ACL configured on the device), then the port is still authenticated, but no filter is dynamically applied to it.
- If the Vendor-Specific attribute specifies the syntax for a filter, but there are insufficient system resources to implement the filter, then the port is still authenticated, but the filter specified in the Vendor-Specific attribute is not applied to the port.

By default, strict security mode is enabled for all 802.1X-enabled interfaces, but you can manually disable or enable it, either globally or for specific interfaces.

Disabling strict security mode globally

To disable strict security mode globally, enter the following commands.

```
Brocade(config)# dot1x-enable
Brocade(config-dot1x)# no global-filter-strict-security
```

After you globally disable strict security mode, you can re-enable it by entering the following command.

```
Brocade(config-dot1x)# global-filter-strict-security
```

Syntax: [no] **global-filter-strict-security**

To disable strict security mode for a specific interface, enter commands such as the following.

```
Brocade(config)# interface e 1/1/1
Brocade(config-if-e10000-1/1/1)# dot1x disable-filter-strict-security
```

To re-enable strict security mode for an interface, enter the following command.

```
Brocade(config-if-e10000-1/1/1)# no dot1x disable-filter-strict-security
```

Syntax: [no] **dot1x disable-filter-strict-security**

The output of the **show dot1x** and **show dot1x config** commands has been enhanced to indicate whether strict security mode is enabled or disabled globally and on an interface. Refer to "Displaying the status of strict security mode" on page 190.

Dynamically applying existing ACLs or MAC address filters

When a port is authenticated using 802.1X security, an IP ACL or MAC address filter that exists in the running-config on the Brocade device can be dynamically applied to the port. To do this, you configure the Filter-ID (type 11) attribute on the RADIUS server. The Filter-ID attribute specifies the name or number of the Brocade IP ACL or MAC address filter.

The following is the syntax for configuring the Filter-ID attribute to refer to a Brocade IP ACL or MAC address filter.

TABLE 27 Configuring Filer-ID attribute

Value	Description
ip.*number*.in	Applies the specified numbered ACL to the 802.1X authenticated port in the inbound direction.
ip.*name*.in	Applies the specified named ACL to the 802.1X authenticated port in the inbound direction.
mac.*number*.in	Applies the specified numbered MAC address filter to the 802.1X authenticated port in the inbound direction.

The following table lists examples of values you can assign to the Filter-ID attribute on the RADIUS server to refer to IP ACLs and MAC address filters configured on a Brocade device.

TABLE 28 IP ACLs and MAC address filters

Possible values for the filter ID attribute on the RADIUS server	ACL or MAC address filter configured on the Brocade device
ip.2.in	access-list 2 permit host 36.48.0.3 access-list 2 permit 36.0.0.0 0.255.255.255
ip.102.in	access-list 102 permit ip 36.0.0.0 0.255.255.255 any
ip.fdry_filter.in	ip access-list standard fdry_filter permit host 36.48.0.3
mac.2.in	mac filter 2 permit 3333.3333.3333 ffff.ffff.ffff any etype eq 0800
mac.2.in mac.3.in	mac filter 2 permit 3333.3333.3333 ffff.ffff.ffff any etype eq 0800 mac filter 3 permit 2222.2222.2222 ffff.ffff.ffff any etype eq 0800

Notes for dynamically applying ACLs or MAC address filters

- The *name* in the Filter ID attribute is case-sensitive.
- You can specify only numbered MAC address filters in the Filter ID attribute. Named MAC address filters are not supported.
- Dynamic ACL filters are supported only for the inbound direction. Dynamic outbound ACL filters are not supported.
- MAC address filters are supported only for the inbound direction. Outbound MAC address filters are not supported.
- Dynamically assigned IP ACLs and MAC address filters are subject to the same configuration restrictions as non-dynamically assigned IP ACLs and MAC address filters.

Configuring per-user IP ACLs or MAC address filters

Per-user IP ACLs and MAC address filters make use of the Vendor-Specific (type 26) attribute to dynamically apply filters to ports. Defined in the Vendor-Specific attribute are Brocade ACL or MAC address filter statements. When the RADIUS server returns the Access-Accept message granting a client access to the network, the Brocade device reads the statements in the Vendor-Specific attribute and applies these IP ACLs or MAC address filters to the client port. When the client disconnects from the network, the dynamically applied filters are no longer applied to the port. If any filters had been applied to the port previous to the client connecting, then those filters are reapplied to the port.

NOTE
Dynamic IP ACL filters and MAC address filters are not supported on the same port at the same time.

The following table shows the syntax for configuring the Brocade Vendor-Specific attributes with ACL or MAC address filter statements.

TABLE 29 Configuring the Brocade Vendor-Specific attributes

Value	Description
ipACL.e.in=<*extended-ACL-entries*>	Applies the specified extended ACL entries to the 802.1X authenticated port in the inbound direction.
macfilter.in=<*mac-filter-entries*>	Applies the specified MAC address filter entries to the 802.1X authenticated port in the inbound direction.

The following table shows examples of IP ACLs and MAC address filters configured in the Brocade Vendor-Specific attribute on a RADIUS server. These IP ACLs and MAC address filters follow the same syntax as other Brocade ACLs and MAC address filters. Refer to the related chapters in this book for information on syntax.

TABLE 30 IP ACLs and MAC address filters

ACL or MAC address filter	Vendor-specific attribute on RADIUS server
MAC address filter with one entry	macfilter.in= deny any any
MAC address filter with two entries	macfilter.in= permit 0000.0000.3333 ffff.ffff.0000 any, macfilter.in= permit 0000.0000.4444 ffff.ffff.0000 any

The RADIUS server allows one instance of the Vendor-Specific attribute to be sent in an Access-Accept message.

Enabling 802.1X port security

By default, 802.1X port security is disabled on Brocade devices. To enable the feature on the device and enter the dot1x configuration level, enter the following command.

```
Brocade(config)# dot1x-enable
Brocade(config-dot1x)#
```

Syntax: [no] **dot1x-enable**

At the dot1x configuration level, you can enable 802.1X port security on all interfaces at once, on individual interfaces, or on a range of interfaces.

For example, to enable 802.1X port security on all interfaces on the device, enter the following command.

```
Brocade(config-dot1x)# enable all
```

Syntax: [no] **enable all**

To enable 802.1X port security on interface 1/3/11, enter the following command.

```
Brocade(config-dot1x)# enable ethernet 1/3/11
```

Syntax: [no] **enable ethernet** *port*

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

To enable 802.1X port security on interfaces 1/3/11 through 1/3/16, enter the following command.

```
Brocade(config-dot1x)# enable ethernet 1/3/11 to 1/3/16
```

Syntax: [no] **enable ethernet** *port* **to** *port*

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

Setting the port control

To activate authentication on an 802.1X-enabled interface, you specify the kind of ***port control*** to be used on the interface. An interface used with 802.1X port security has two virtual access points: a controlled port and an uncontrolled port:

- The controlled port can be either the authorized or unauthorized state. In the authorized state, it allows normal traffic to pass between the Client and the Authenticator. In the unauthorized state, no traffic is allowed to pass.
- The uncontrolled port allows only EAPOL traffic between the Client and the Authentication Server.

Refer to Figure 3 for an illustration of this concept.

By default, all controlled ports on the device are in the authorized state, allowing all traffic. When you activate authentication on an 802.1X-enabled interface, its controlled port is placed in the unauthorized state. When a Client connected to the interface is successfully authenticated, the controlled port is then placed in the authorized state. The controlled port remains in the authorized state until the Client logs off.

To activate authentication on an 802.1X-enabled interface, you configure the interface to place its controlled port in the authorized state when a Client is authenticated by an Authentication Server. To do this, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# dot1x port-control auto
```

Syntax: [**no**] **dot1x port-control** [**force-authorized** | **force-unauthorized** | **auto**]

When an interface control type is set to **auto**, the controlled port is initially set to unauthorized, but is changed to authorized when the connecting Client is successfully authenticated by an Authentication Server.

The port control type can be one of the following:

force-authorized – The controlled port is placed unconditionally in the authorized state, allowing all traffic. This is the default state for ports on the Brocade device.

force-unauthorized – The controlled port is placed unconditionally in the unauthorized state.

auto – The controlled port is unauthorized until authentication takes place between the Client and Authentication Server. Once the Client passes authentication, the port becomes authorized. This activates authentication on an 802.1X-enabled interface.

NOTE
You cannot enable 802.1X port security on ports that have any of the following features enabled:

- Link aggregation
- Metro Ring Protocol (MRP)
- Mirror port
- Trunk port

Configuring periodic re-authentication

You can configure the device to periodically re-authenticate Clients connected to 802.1X-enabled interfaces. When you enable periodic re-authentication, the device re-authenticates Clients every 3,600 seconds by default. You can optionally specify a different re-authentication interval of between 1 – 4294967295 seconds.

To configure periodic re-authentication using the default interval of 3,600 seconds, enter the following command.

```
Brocade(config-dot1x)# re-authentication
```

Syntax: [**no**] **re-authentication**

To configure periodic re-authentication with an interval of 2,000 seconds, enter the following commands.

```
Brocade(config-dot1x)# re-authentication
Brocade(config-dot1x)# timeout re-authperiod 2000
```

Syntax: [**no**] **timeout re-authperiod** *seconds*

The re-authentication interval is a global setting, applicable to all 802.1X-enabled interfaces. To re-authenticate Clients connected to a specific port manually, use the **dot1x re-authenticate** command. Refer to "Re-authenticating a port manually", below.

Re-authenticating a port manually

When periodic re-authentication is enabled, by default the Brocade device re-authenticates Clients connected to an 802.1X-enabled interface every 3,600 seconds (or the time specified by the **dot1x timeout re-authperiod** command). You can also manually re-authenticate Clients connected to a specific port.

For example, to re-authenticate Clients connected to interface 1/3/1, enter the following command.

```
Brocade# dot1x re-authenticate e 1/3/1
```

Syntax: dot1x re-authenticate ethernet *port*

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

Setting the quiet period

If the Brocade device is unable to authenticate the Client, the Brocade device waits a specified amount of time before trying again. The amount of time the Brocade device waits is specified with the **quiet-period** parameter. The **quiet-period** parameter can be from 1 – 4294967295 seconds. The default is 60 seconds.

For example, to set the quiet period to 30 seconds, enter the following command.

```
Brocade(config-dot1x)# timeout quiet-period 30
```

Syntax: [**no**] **timeout quiet-period** *seconds*

Specifying the wait interval and number of EAP-request/ identity frame retransmissions from the Brocade device

When the Brocade device sends an EAP-request/identity frame to a Client, it expects to receive an EAP-response/identity frame from the Client. By default, if the Brocade device does not receive an EAP-response/identity frame from a Client, the device waits 30 seconds, then retransmits the EAP-request/identity frame. Also by default, the Brocade device retransmits the EAP-request/identity frame a maximum of two times. You can optionally configure the amount of time the device will wait before retransmitting an EAP-request/identity frame, and the number of times the EAP-request/identity frame will be transmitted. This section provides the command syntax for these features.

Setting the wait interval for EAP frame retransmissions

By default, if the Brocade device does not receive an EAP-response/identity frame from a Client, the device waits 30 seconds, then retransmits the EAP-request/identity frame. You can optionally change the amount of time the Brocade device waits before retransmitting the EAP-request/identity frame to the Client.

For example, to cause the Brocade device to wait 60 seconds before retransmitting an EAP-request/identity frame to a Client, enter the following command.

```
Brocade(config-dot1x)# timeout tx-period 60
```

If the Client does not send back an EAP-response/identity frame within 60 seconds, the device will transmit another EAP-request/identity frame.

Syntax: [**no**] **timeout tx-period** *seconds*

where *seconds* is a value from 1–4294967295. The default is 30 seconds.

Setting the maximum number of EAP frame retransmissions

The Brocade device retransmits the EAP-request/identity frame a maximum of two times. If no EAP-response/identity frame is received from the Client after two EAP-request/identity frame retransmissions (or the amount of time specified with the **auth-max** command), the device restarts the authentication process with the Client.

You can optionally change the number of times the Brocade device should retransmit the EAP-request/identity frame. You can specify between 1 – 10 frame retransmissions. For example, to configure the device to retransmit an EAP-request/identity frame to a Client a maximum of three times, enter the following command:

```
Brocade(config-dot1x)# auth-max 3
```

Syntax: **auth-max** *value*

value is a number from 1–10. The default is 2.

Wait interval and number of EAP-request/ identity frame retransmissions from the RADIUS server

Acting as an intermediary between the RADIUS Authentication Server and the Client, the Brocade device receives RADIUS messages from the RADIUS server, encapsulates them as EAPOL frames, and sends them to the Client. By default, when the Brocade device relays an EAP-Request frame from the RADIUS server to the Client, it expects to receive a response from the Client within 30 seconds. If the Client does not respond within the allotted time, the device retransmits the EAP-Request frame to the Client. Also by default, the Brocade device retransmits the EAP-request frame twice. If no EAP-response frame is received from the Client after two EAP-request frame retransmissions, the device restarts the authentication process with the Client.

You can optionally configure the amount of time the device will wait before retransmitting an EAP-request/identity frame, and the number of times the EAP-request/identity frame will be transmitted. This section provides the command syntax for these features.

Setting the wait interval for EAP frame retransmissions

By default, when the Brocade device relays an EAP-Request frame from the RADIUS server to the Client, it expects to receive a response from the Client within 30 seconds. You can optionally specify the wait interval using the **supptimeout** command.

For example, to configure the device to retransmit an EAP-Request frame if the Client does not respond within 45 seconds, enter the following command.

```
Brocade(config-dot1x)# supptimeout 45
```

Syntax: supptimeout *seconds*

seconds is a number from 1–4294967295 seconds. The default is 30 seconds.

Setting the maximum number of EAP frame retransmissions

You can optionally specify the number of times the Brocade device will retransmit the EAP-request frame. You can specify between 1–10 frame retransmissions. For example, to configure the device to retransmit an EAP-request frame to a Client a maximum of three times, enter the following command.

```
Brocade(config-dot1x)# maxreq 3
```

Syntax: maxreq *value*

value is a number from 1–10. The default is 2.

Specifying a timeout for retransmission of messages to the authentication server

When performing authentication, the Brocade device receives EAPOL frames from the Client and passes the messages on to the RADIUS server. The device expects a response from the RADIUS server within 30 seconds. If the RADIUS server does not send a response within 30 seconds, the Brocade device retransmits the message to the RADIUS server. The time constraint for retransmission of messages to the Authentication Server can be between 0 – 4294967295 seconds.

For example, to configure the device to retransmit a message if the Authentication Server does not respond within 45 seconds, enter the following command.

```
Brocade(config-dot1x)# servertimeout 45
```

Syntax: servertimeout *seconds*

Initializing 802.1X on a port

To initialize 802.1X port security on a port, enter a command such as the following.

```
Brocade# dot1x initialize e 1/3/1
```

Syntax: dot1x initialize ethernet *port*

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

Allowing access to multiple hosts

Brocade devices support 802.1X authentication for ports with more than one host connected to them. If there are multiple hosts connected to a single 802.1X-enabled port, the Brocade device authenticates each of them individually. Refer to "Configuring 802.1X multiple-host authentication" on page 179.

Configuring 802.1X multiple-host authentication

When multiple hosts are connected to the same 802.1X-enabled port, the functionality described in "How 802.1X multiple-host authentication works" on page 160 is enabled by default. You can optionally do the following:

- Specify the authentication-failure action
- Specify the number of authentication attempts the device makes before dropping packets
- Disabling aging for dot1x-mac-sessions
- Configure aging time for blocked clients
- Moving native VLAN mac-sesions to restrict VLAN
- Clear the dot1x-mac-session for a MAC address

Specifying the authentication-failure action

In an 802.1X multiple-host configuration, if RADIUS authentication for a client is unsuccessful, either traffic from that client is dropped in hardware (the default), or the client port is placed in a "restricted" VLAN. You can specify which of these authentication-failure actions to use. When you enable 802.1X, the default authentication-failure action is to drop client traffic.

If you configure the authentication-failure action to place the client port in a restricted VLAN, you can specify the ID of the restricted VLAN. If you do not specify a VLAN ID, the default VLAN is used.

You can configure the authentication-failure action using one of the following methods:

- Configure the same authentication-failure action for all ports on the device (globally).
- Configure an authentication-failure action on individual ports.

NOTE
You cannot configure the authentication-failure action globally and per-port at the same time.

To configure the authentication-failure action for all ports on the device to place the client port in a restricted VLAN, enter the following commands.

```
Brocade(config)# dot1x-enable
Brocade(config-dot1x)# auth-fail-action restricted-vlan
```

Syntax: [no] **auth-fail-action restricted-vlan**

To specify VLAN 300 as the restricted VLAN for all ports on the device, enter the **auth-fail-vlanid** *num* command.

```
Brocade(config-dot1x)# auth-fail-vlanid 300
```

Syntax: [no] **auth-fail-vlanid** *vlan-id*

To specify on an individual port that the authentication-failure action is to place the client port in restricted VLAN 300, enter the following command at the interface configuration level.

```
Brocade(config-if-e10000-1/1/1)# dot1x auth-fail-action restrict-vlan 300
```

Syntax: [no] **dot1x auth-fail-action restrict-vlan** *vlan-id*

Specifying the number of authentication attempts the device makes before dropping packets

When the authentication-failure action is to drop traffic from the Client, and the initial authentication attempt made by the device to authenticate the Client is unsuccessful, the Brocade device immediately retries to authenticate the Client. After three unsuccessful authentication attempts, the Client dot1x-mac-session is set to “access-denied”, causing traffic from the Client to be dropped in hardware.

Optionally, you can configure the number of authentication attempts the device makes before dropping traffic from the Client. To do so, enter a command such as the following.

```
Brocade(config-dot1x)# auth-fail-max-attempts 2
```

Syntax: [no] **auth-fail-max-attempts** *attempts*

By default, the device makes three attempts to authenticate a Client before dropping packets from the Client. You can specify from 1 through 10 authentication attempts.

Disabling aging for dot1x-mac-sessions

The dot1x-mac-sessions for Clients authenticated or denied by a RADIUS server are aged out if no traffic is received from the Client MAC address for a certain period of time. After a Client dot1x-mac-session is aged out, the Client must be re-authenticated:

- ***Permitted*** dot1x-mac-sessions, which are the dot1x-mac-sessions for authenticated Clients, as well as for non-authenticated Clients whose ports have been placed in the restricted VLAN, are aged out if no traffic is received from the Client MAC address over the normal MAC aging interval on the Brocade device.
- ***Denied*** dot1x-mac-sessions, which are the dot1x-mac-sessions for non-authenticated Clients that are blocked by the Brocade device are aged out over a configurable software aging period. (Refer to the next section for more information on configuring the software aging period).

You can optionally disable aging of the permitted or denied dot1x-mac-sessions, or both, on the Brocade device.

To disable aging of the permitted dot1x-mac-sessions, enter the following command.

```
Brocade(config-dot1x)# mac-session-aging no-aging permitted-mac-only
```

Syntax: [no] **mac-session-aging no-aging permitted-mac-only**

To disable aging of the denied dot1x-mac-sessions, enter the following command.

```
Brocade(config-dot1x)# mac-session-aging no-aging denied-mac-only
```

Syntax: [no] **mac-session-aging no-aging denied-mac-only**

NOTE
This command enables aging of permitted sessions.

As a shortcut, use the command [no] **mac-session-aging** to enable or disable aging for permitted and denied sessions.

Specifying the aging time for blocked clients

When the Brocade device is configured to drop traffic from non-authenticated Clients, traffic from the blocked Clients is dropped in hardware, without being sent to the CPU. A Layer 2 CAM entry is created that drops traffic from the blocked Client MAC address in hardware. If no traffic is received from the blocked Client MAC address for a certain amount of time, this Layer 2 CAM entry is aged out. If traffic is subsequently received from the Client MAC address, then an attempt can be made to authenticate the Client again.

Aging of the Layer 2 CAM entry for a blocked Client MAC address occurs in two phases, known as ***hardware aging*** and ***software aging***. The hardware aging period is fixed at 70 seconds and is non-configurable. The software aging time is configurable through the CLI.

Once the Brocade device stops receiving traffic from a blocked Client MAC address, the hardware aging begins and lasts for a fixed period of time. After the hardware aging period ends, the software aging period begins. The software aging period lasts for a configurable amount of time (by default 120 seconds). After the software aging period ends, the blocked Client MAC address ages out, and can be authenticated again if the Brocade device receives traffic from the Client MAC address.

Change the length of the software aging period for a blocked Client MAC address by entering the **mac-age-time** *num* command.

```
Brocade(config)# mac-age-time 180
```

Syntax: [no] **mac-age-time** *seconds*

You can specify from 1–65535 seconds. The default is 120 seconds.

Moving native VLAN mac-sesions to restrict VLAN

You can move the native VLAN mac-sessions to restrict VLAN on authentication failure. You can configure the option of overriding the dual-mode port native untagged VLAN with restricted VLAN in case 802.1x authentication fails and there is no RADIUS assigned VLAN. Use this command when you configure multi-device port authentication and 802.1X authentication configuration with dynamic VLAN assignment from RADIUS Server on the same port.

Example

```
Brocade(config-dot1x)# auth-fail-force-restrict
```

[no] **auth-fail-force-restrict**

Clearing a dot1x-mac-session for a MAC address

You can clear the dot1x-mac-session for a specified MAC address, so that the Client with that MAC address can be re-authenticated by the RADIUS server.

Example

```
Brocade# clear dot1x mac-session 0000.0034.abd4
```

Syntax: clear dot1x mac-session *mac-address*

MAC address filters for EAP frames

You can create MAC address filters to permit or deny EAP frames. To do this, you specify the Brocade device 802.1X group MAC address as the destination address in a MAC address filter, then apply the filter to an interface.

Creating MAC address filters for EAP on most devices

For example, the following command creates a MAC address filter that denies frames with the destination MAC address of 0000.00c2.0003, which is the 802.1X group MAC address on the Brocade device.

```
Brocade(config)# mac filter 1 deny any 0000.00c2.0003 ffff.ffff.ffff
```

The following commands apply this filter to interface e1/ 3/1.

```
Brocade(config)# interface e 1/3/11
Brocade(config-if-e10000-1/3/1)# mac filter-group 1
```

Refer to "Defining MAC address filters" on page 239 for more information.

Configuring VLAN access for non-EAP-capable clients

You can configure the Brocade device to grant "guest" or restricted VLAN access to clients that do not support Extensible EAP. The restricted VLAN limits access to the network or applications, instead of blocking access to these services altogether.

When the Brocade device receives the first packet (non-EAP packet) from a client, the device waits for 10 seconds or the amount of time specified with the **timeout restrict-fwd-period** command. If the Brocade device does not receive subsequent packets after the timeout period, the device places the client on the restricted VLAN.

This feature is disabled by default. To enable this feature and change the timeout period, enter commands such as the following.

```
Brocade(config)# dot1x-enable
Brocade(config-dot1x)# restrict-forward-non-dot1x
Brocade(config-dot1x)# timeout restrict-fwd-period 15
```

Once the *success* timeout action is enabled, use the **no** form of the command to reset the RADIUS timeout behavior to *retry*.

Syntax: timeout restrict-fwd-period *num*

The *num* parameter is a value from 0 to 4294967295. The default value is 10.

802.1X accounting configuration

802.1X accounting enables the recording of information about 802.1X clients who were successfully authenticated and allowed access to the network. When 802.1X accounting is enabled on the Brocade device, it sends the following information to a RADIUS server whenever an authenticated 802.1X client (user) logs into or out of the Brocade device:

- The user name
- The session ID

- The user MAC address
- The authenticating physical port number

An Accounting Start packet is sent to the RADIUS server when a user is successfully authenticated. The Start packet indicates the start of a new session and contains the user MAC address and physical port number. The 802.1X session state will change to Authenticated and Permit after receiving a response from the accounting server for the accounting Start packet. If the Accounting service is not available, the 802.1X session status will change to Authenticated and Permit after a RADIUS timeout. The device will retry authentication requests three times (the default), or the number of times configured on the device.

An ***Accounting Stop packet*** is sent to the RADIUS server when one of the following events occur:

- The user logs off
- The port goes down
- The port is disabled
- The user fails to re-authenticate after a RADIUS timeout
- The 802.1X port control-auto configuration changes
- The MAC session clears (through use of the **clear dot1x mac-session** CLI command)

The Accounting Stop packet indicates the end of the session and the time the user logged out.

802.1X accounting attributes for RADIUS

Brocade devices support the following RADIUS attributes for 802.1X accounting.

TABLE 31 802.1X accounting attributes for RADIUS

Attribute name	Attribute ID	Data Type	Description
Acct-Session-ID	44	Integer	The account session ID, which is a number from 1 to 4294967295.
Acct-Status-Type	40	integer	Indicates whether the accounting request marks the beginning (start) or end (stop) of the user service. 1 – Start 2 – Stop
Calling-Station-Id	31	string	The supplicant MAC address in ASCII format (upper case only), with octet values separated by a dash (-). For example 00-10-A4-23-19-C0
NAS-Port	5	integer	The physical port number.
NAS-Port-Type	61	integer	The physical port type.
user-name	1	string	The user name.

Enabling 802.1X accounting

To enable 802.1X accounting, enter the following command.

```
Brocade(config)# aaa accounting dot1x default start-stop radius none
```

Syntax: aaa accounting dot1x default start-stop radius | none

radius – Use the list of all RADIUS servers that support 802.1X for authentication.

none – Use no authentication. The client is automatically authenticated without the device using information supplied by the client.

NOTE
If you specify both **radius** and **none**, make sure **radius** comes before **none**.

Displaying 802.1X information

You can display the following 802.1X-related information:

- The 802.1X configuration on the device and on individual ports
- Statistics about the EAPOL frames passing through the device
- 802.1X-enabled ports dynamically assigned to a VLAN
- User-defined and dynamically applied MAC address filters and IP ACLs currently active on the device
- The 802.1X multiple-host configuration

Displaying 802.1X configuration information

To display information about the 802.1X configuration on the Brocade device, enter the **show dot1x** command.

```
Brocade# show dot1x
PAE Capability:    Authenticator Only
system-auth-control: Enable
re-authentication: Disable
global-filter-strict-security: Enable
quiet-period:    60 Seconds
tx-period:    30 Seconds
supptimeout:    30 Seconds
servertimeout:     30 Seconds
maxreq:    2
re-authperiod:    3600 Seconds
Protocol Version:    1
```

Syntax: show dot1x

The following table describes the information displayed by the **show dot1x** command.

TABLE 32 Output from the **show dot1x command**

Field	Description
PAE Capability	The Port Access Entity (PAE) role for the Brocade device. This is always "Authenticator Only".
system-auth-control	Whether system authentication control is enabled on the device. The **dot1x-enable** command enables system authentication control on the device.
re-authentication	Whether periodic re-authentication is enabled on the device. Refer to "Configuring periodic re-authentication" on page 175. When periodic re-authentication is enabled, the device automatically re-authenticates Clients every 3,600 seconds by default.

TABLE 32 Output from the **show dot1x command** (Continued)

Field	Description
global-filter-strict-security	Whether strict security mode is enabled or disabled globally. Refer to "Disabling and enabling strict security mode for dynamic filter assignment" on page 171.
quiet-period	When the Brocade device is unable to authenticate a Client, the amount of time the Brocade device waits before trying again (default 60 seconds). Refer to "Setting the quiet period" on page 176 for information on how to change this setting.
tx-period	When a Client does not send back an EAP-response/identity frame, the amount of time the Brocade device waits before retransmitting the EAP-request/identity frame to a Client (default 30 seconds). Refer to "Setting the wait interval for EAP frame retransmissions" on page 176 for information on how to change this setting.
supp-timeout	When a Client does not respond to an EAP-request frame, the amount of time before the Brocade device retransmits the frame. Refer to "Setting the wait interval for EAP frame retransmissions" on page 177 for information on how to change this setting.
server-timeout	When the Authentication Server does not respond to a message sent from the Client, the amount of time before the Brocade device retransmits the message. Refer to "Specifying a timeout for retransmission of messages to the authentication server" on page 178 for information on how to change this setting.
maxreq	The number of times the Brocade device retransmits an EAP-request/identity frame if it does not receive an EAP-response/identity frame from a Client (default 2 times). Refer to "Setting the maximum number of EAP frame retransmissions" on page 177 for information on how to change this setting.
re-authperiod	How often the device automatically re-authenticates Clients when periodic re-authentication is enabled (default 3,600 seconds). Refer to "Configuring periodic re-authentication" on page 175 for information on how to change this setting.
Protocol Version	The version of the 802.1X protocol in use on the device.

To display information about the 802.1X configuration on an individual port, enter the **show dot1x configuration ethernet** command.

```
Brocade# show dot1x configuration ethernet 1/1/3
Port-Control                    : control-auto
filter strict security          : Enable
Action on RADIUS timeout        : Treat as a failed authentication
  re-authenticate               : 150 seconds
PVID State                      : Normal (101)
Original PVID                   : 101
PVID mac total                  : 1
PVID mac authorized             : 1
num mac sessions                : 1
num mac authorized              : 1
Number of Auth filter           : 0
```

Syntax: show dot1x config ethernet *port*

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

The following additional information is displayed in the **show dot1x config** command for an interface.

TABLE 33 Output from the **show dot1x config** command for an interface

Field	Description
Authenticator PAE state	The current status of the Authenticator PAE state machine. This can be INITIALIZE, DISCONNECTED, CONNECTING, AUTHENTICATING, AUTHENTICATED, ABORTING, HELD, FORCE_AUTH, or FORCE_UNAUTH. **NOTE:** When the Authenticator PAE state machine is in the AUTHENTICATING state, if the reAuthenticate, eapStart, eapLogoff, or authTimeout parameters are set to TRUE, it may place the Authenticator PAE state machine indefinitely in the ABORTING state. If this should happen, use the **dot1x initialize** command to initialize 802.1X port security on the port, or unplug the Client or hub connected to the port, then reconnect it.
Backend Authentication state	The current status of the Backend Authentication state machine. This can be REQUEST, RESPONSE, SUCCESS, FAIL, TIMEOUT, IDLE, or INITIALIZE.
AdminControlledDirections	Indicates whether an unauthorized controlled port exerts control over communication in both directions (disabling both reception of incoming frames and transmission of outgoing frames), or just in the incoming direction (disabling only reception of incoming frames). On Brocade devices, this parameter is set to BOTH.
OperControlledDirections	The setting for the OperControlledDirections parameter, as defined in the 802.1X standard. According to the 802.1X standard, if the AdminControlledDirections parameter is set to BOTH, the OperControlledDirections parameter is unconditionally set to BOTH. Since the AdminControlledDirections parameter on Brocade devices is always set to BOTH, the OperControlledDirections parameter is also set to BOTH.
AuthControlledPortControl	The port control type configured for the interface. If set to auto, authentication is activated on the 802.1X-enabled interface.
AuthControlledPortStatus	The current status of the interface controlled port either authorized or unauthorized.
multiple-hosts	Whether the port is configured to allow multiple Supplicants accessing the interface on the Brocade device through a hub. Refer to "Allowing access to multiple hosts" on page 179 for information on how to change this setting.

Displaying 802.1X statistics

To display 802.1X statistics for an individual port, enter the **show dot1x statistics** command.

```
Brocade# show dot1x statistics e 1/3/3

Port 1/3/3 Statistics:
RX EAPOL Start:    0
RX EAPOL Logoff:    0
RX EAPOL Invalid:    0
RX EAPOL Total:    0
RX EAP Resp/Id:    0
RX EAP Resp other than Resp/Id:    0
RX EAP Length Error:    0
Last EAPOL Version:    0
Last EAPOL Source:    0007.9550.0B83
TX EAPOL Total:    217
TX EAP Req/Id:    163
TX EAP Req other than Req/Id:    0
```

Syntax: show dot1x statistics ethernet *port*

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

The following table describes the information displayed by the **show dot1x statistics** command for an interface.

TABLE 34 Output from the **show dot1x statistics** command

Field	Statistics
RX EAPOL Start	The number of EAPOL-Start frames received on the port.
RX EAPOL Logoff	The number of EAPOL-Logoff frames received on the port.
RX EAPOL Invalid	The number of invalid EAPOL frames received on the port.
RX EAPOL Total	The total number of EAPOL frames received on the port.
RX EAP Resp/Id	The number of EAP-Response/Identity frames received on the port
RX EAP Resp other than Resp/Id	The total number of EAPOL-Response frames received on the port that were not EAP-Response/Identity frames.
RX EAP Length Error	The number of EAPOL frames received on the port that have an invalid packet body length.
Last EAPOL Version	The version number of the last EAPOL frame received on the port.
Last EAPOL Source	The source MAC address in the last EAPOL frame received on the port.
TX EAPOL Total	The total number of EAPOL frames transmitted on the port.
TX EAP Req/Id	The number of EAP-Request/Identity frames transmitted on the port.
TX EAP Req other than Req/Id	The number of EAP-Request frames transmitted on the port that were not EAP-Request/Identity frames.

Clearing 802.1X statistics

You can clear the 802.1X statistics counters on all interfaces at once, on individual interfaces, or on a range of interfaces.

For example, to clear the 802.1X statistics counters on all interfaces on the device, enter the **clear dot1x statistics all** command.

```
Brocade# clear dot1x statistics all
```

Syntax: clear dot1x statistics all

To clear the 802.1X statistics counters on interface e 1/3/11, enter the following command.

```
Brocade# clear dot1x statistics e 1/3/11
```

Syntax: clear dot1x statistics ethernet *port*

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

Displaying dynamically assigned VLAN information

The **show interface** command displays the VLAN to which an 802.1X-enabled port has been dynamically assigned, as well as the port from which it was moved (that is, the port default VLAN).

The following example of the **show interface** command indicates the port dynamically assigned VLAN. Information about the dynamically assigned VLAN is shown in bold type.

```
Brocade# show interface ethernet 1/2/2
Ethernet1/2/2 is up, line protocol is up
  Hardware is FastEthernet, address is 0000.00a0.4681 (bia 000.00a0.4681)
  Configured speed auto, actual 100Mbit, configured duplex fdx, actual fdx
  Member of L2 VLAN ID 2 (dot1x-RADIUS assigned), original L2 VLAN ID is 1,
  port is untagged, port state is FORWARDING
  STP configured to ON, priority is level0, flow control enabled
  mirror disabled, monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  MTU 1518 bytes
  300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 256 bits/sec, 0 packets/sec, 0.00% utilization
  3 packets input, 192 bytes, 0 no buffer
  Received 0 broadcasts, 0 multicasts, 3 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runts, 0 giants, DMA received 3 packets
  919 packets output, 58816 bytes, 0 underruns
  Transmitted 1 broadcasts, 916 multicasts, 2 unicasts
  0 output errors, 0 collisions, DMA transmitted 919 packets
```

In this example, the 802.1X-enabled port has been moved from VLAN 1 to VLAN 2. When the client disconnects, the port will be moved back to VLAN 1.

The **show run** command also indicates the VLAN to which the port has been dynamically assigned. When you enter the **show run** command, the output indicates that the port is a member of the VLAN to which it was dynamically assigned through 802.1X. If you then enter the **write memory** command, the VLAN to which the port is currently assigned becomes the port default VLAN in the device configuration.

If the VLAN name supplied by the RADIUS server corresponds to a statically configured VLAN, the output indicates that the port is a member of the VLAN to which it was dynamically assigned through 802.1X. If you then enter the **write memory** command, the VLAN to which the port is currently assigned becomes the port default VLAN in the device configuration.

Displaying information about dynamically applied MAC address filters and IP ACLs

You can display information about currently active user-defined and dynamically applied MAC address filters and IP ACLs.

Displaying user-defined MAC address filters and IP ACLs

To display the user-defined MAC address filters active on the device, enter the following command.

```
Brocade# show dot1x mac-address filter

Port 1/1/3 (User defined MAC Address Filter) :
      mac filter 1 permit any any
```

Syntax: show dot1x mac-address-filter

To display the user-defined IP ACLs active on the device, enter the **show dot1x ip-ACL** command.

```
Brocade# show dot1x ip-ACL

Port 1/1/3 (User defined IP ACLs):

Extended IP access list Port_1/1/3_E_IN
permit udp any any

Extended IP access list Port_1/1/3_E_OUT
permit udp any any
```

Syntax: show dot1x ip-ACL

Displaying dynamically applied MAC address filters and IP ACLs

To display the dynamically applied MAC address filters active on an interface, enter a command such as the following.

```
Brocade# show dot1x mac-address-filter e 1/1/3

Port 1/1/3 MAC Address Filter information:
  802.1X Dynamic MAC Address Filter :
     mac filter-group 2
  Port default MAC Address Filter:
     No mac address filter is set
```

Syntax: show dot1x mac-address-filter all | ethernet *port*

The **all** keyword displays all dynamically applied MAC address filters active on the device.

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

To display the dynamically applied IP ACLs active on an interface, enter a command such as the following.

```
Brocade# show dot1x ip-ACL ethernet 1/1/3

Port 1/1/3 IP ACL information:
  802.1X dynamic IP ACL (user defined) in:
    ip access-list extended Port_1/1/3_E_IN in
  Port default IP ACL in:
    No inbound ip access-list is set
  802.1X dynamic IP ACL (user defined) out:
    ip access-list extended Port_1/1/3_E_OUT out
  Port default IP ACL out:
    No outbound ip access-list is set
```

Syntax: show dot1x ip-ACL all | ethernet *port*

The **all** keyword displays all dynamically applied IP ACLs active on the device.

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

Displaying the status of strict security mode

The output of the **show dot1x** and **show dot1x config** commands indicate whether strict security mode is enabled or disabled globally and on an interface.

Displaying the status of strict security mode globally on the device

To display the status of strict security mode globally on the device, enter the **show dot1x** command.

```
Brocade# show dot1x
PAE Capability:    Authenticator Only
system-auth-control: Enable
re-authentication: Disable
global-filter-strict-security: Enable
quiet-period:    60 Seconds
tx-period:    30 Seconds
supptimeout:    30 Seconds
servertimeout:     30 Seconds
maxreq:    2
re-authperiod:    3600 Seconds
security-hold-time: 60 Seconds
Protocol Version:    1
```

Syntax: show dot1x

Displaying the status of strict security mode on an interface

To display the status of strict security mode on an interface, enter a command such as the following

```
Brocade# show dot1x config e 1/1/3

Port 1/1/3 Configuration:
Authenticator PAE state:    AUTHENTICATED
Backend Authentication state:     IDLE
AdminControlledDirections:    BOTH
OperControlledDirections:    BOTH
AuthControlledPortControl:     Auto
AuthControlledPortStatus:    authorized
quiet-period:    60 Seconds
tx-period:    30 Seconds
supptimeout:    30 Seconds
servertimeout:     30 Seconds
maxreq:    2
re-authperiod:    3600 Seconds
security-hold-time: 60 Seconds
re-authentication: Disable
multiple-hosts: Disable
filter-strict-security: Enable
Protocol Version:    1
```

Syntax: show dot1x config ethernet *port*

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

Displaying 802.1X multiple-host authentication information

You can display the following information about 802.1X multiple-host authentication:

- Information about the 802.1X multiple-host configuration
- The dot1x-mac-sessions on each port
- The number of users connected on each port in a 802.1X multiple-host configuration

Displaying 802.1X multiple-host configuration information

The output of the **show dot1x** and **show dot1x config** commands displays information related to 802.1X multiple-host authentication.

The following is an example of the output of the **show dot1x** command. The information related to multiple-host authentication is highlighted in bold.

```
Brocade# show dot1x

Number of Ports enabled         : 2
Re-Authentication               : Enabled
Authentication-fail-action      : Restricted VLAN
Authentication Failure VLAN     : 111
Mac Session Aging               : Disabled for permitted MAC sessions
Mac Session max-age             : 60 seconds
Protocol Version                : 1
quiet-period                    : 5 Seconds
tx-period                       : 30 Seconds
supptimeout                     : 30 Seconds
servertimeout                   : 30 Seconds
maxreq                          : 2
re-authperiod                   : 3600 Seconds
security-hold-time              : 60 Seconds
re-authentication               : Enable
Flow based multi-user policy      : Disable
```

Syntax: show dot1x

Table 35 describes the bold fields in the display.

TABLE 35 Output from the **show dot1x** command for multiple host authentication

Field	Description
Authentication-fail-action	The configured authentication-failure action. This can be Restricted VLAN or Block Traffic.
Authentication Failure VLAN	If the authentication-failure action is Restricted VLAN, the ID of the VLAN to which unsuccessfully authenticated Client ports are assigned.
Mac Session Aging	Whether aging for dot1x-mac-sessions has been enabled or disabled for permitted or denied dot1x-mac-sessions.
Mac Session max-age	The configured software aging time for dot1x-mac-sessions.
Flow based multi-user policy	The dynamically assigned IP ACLs and MAC address filters used in the 802.1X multiple-host configuration.

The output of the **show dot1x config** command for an interface displays the configured port control for the interface. This command also displays information related to 802.1X multiple host-authentication.

The following is an example of the output of the **show dot1x config** command for an interface.

```
Brocade# show dot1x config e 1/3/1

Port-Control                    : control-auto
filter strict security          : Enable
PVID State                      : Restricted (10)
Original PVID                   : 10
PVID mac total                  : 1
PVID mac authorized             : 0
num mac sessions                : 1
num mac authorized              : 0
```

Syntax: show dot1x config ethernet *port*

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

The following table lists the fields in the display.

TABLE 36 Output from the **show dot1x config** command

Field	Description
Port-Control	The configured port control type for the interface. This can be one of the following: **force-authorized** – The controlled port is placed unconditionally in the authorized state, allowing all traffic. This is the default state for ports on the Brocade device. **force-unauthorized** – The controlled port is placed unconditionally in the unauthorized state. No authentication takes place for any connected 802.1X Clients. **auto** – The authentication status for each 802.1X Client depends on the authentication status returned from the RADIUS server.
filter strict security	Whether strict security mode is enabled or disabled on the interface.
PVID State	The port default VLAN ID (PVID) and the state of the port PVID. The PVID state can be one of the following **Normal** – The port PVID is not set by a RADIUS server, nor is it the restricted VLAN. **RADIUS** – The port PVID was dynamically assigned by a RADIUS server. **RESTRICTED** – The port PVID is the restricted VLAN.
Original PVID	The originally configured (not dynamically assigned) PVID for the port.
PVID mac total	The number of devices transmitting untagged traffic on the port PVID.
PVID mac authorized	The number of devices transmitting untagged traffic on the port PVID as a result of dynamic VLAN assignment.
num mac sessions	The number of dot1x-mac-sessions on the port.
num mac authorized	The number of authorized dot1x-mac-sessions on the port.

Displaying information about the dot1x MAC sessions on each port

The **show dot1x mac-session** command displays information about the dot1x-mac-sessions on each port on the device. The output also shows the authenticator PAE state.

Example

```
Brocade# show dot1x mac-session

Port  MAC/(username)                  Vlan Auth     ACL   Age   PAE
                                           State                 State
-------------------------------------------------------------------------------
1/1/1 0000.0098.24f7 :User            10   permit   none  S20   AUTHENTICATED
```

Syntax: show dot1x mac-session

Table 37 lists the new fields in the display.

TABLE 37 Output from the **show dot1x mac-session** command

Field	Description
Port	The port on which the dot1x-mac-session exists.
MAC/ (username)	The MAC address of the Client and the username used for RADIUS authentication.
Vlan	The VLAN to which the port is currently assigned.
Auth-State	The authentication state of the dot1x-mac-session. This can be one of the following **permit** – The Client has been successfully authenticated, and traffic from the Client is being forwarded normally. **blocked** – Authentication failed for the Client, and traffic from the Client is being dropped in hardware. **restricted** – Authentication failed for the Client, but traffic from the Client is allowed in the restricted VLAN only. **init** - The Client is in is in the process of 802.1X authentication, or has not started the authentication process.
Age	The software age of the dot1x-mac-session.
PAE State	The current status of the Authenticator PAE state machine. This can be INITIALIZE, DISCONNECTED, CONNECTING, AUTHENTICATING, AUTHENTICATED, ABORTING, HELD, FORCE_AUTH, or FORCE_UNAUTH. **NOTE:** When the Authenticator PAE state machine is in the AUTHENTICATING state, if the reAuthenticate, eapStart, eapLogoff, or authTimeout parameters are set to TRUE, it may place the Authenticator PAE state machine indefinitely in the ABORTING state. If this should happen, use the dot1x initialize command to initialize 802.1X port security on the port, or unplug the Client or hub connected to the port, then reconnect it.

Displaying information about the ports in an 802.1X multiple-host configuration

To display information about the ports in an 802.1X multiple-host configuration, enter the **show dot1x mac-session brief** command.

```
Brocade(config-dot1x)# show dot1x mac-session brief
Port            Number of  Number of        Dynamic Dynamic Dynamic
                users      Authorized users VLAN    ACL     MAC-Filt
-------------------------------------------------------------------
1/1/1           0          0                no      no      no
1/1/2           0          0                no      no      no
1/1/3           0          0                no      no      no
1/1/4           0          0                no      no      no
1/1/5           0          0                no      no      no
1/1/6           0          0                no      no      no
1/1/7           0          0                no      no      no
1/1/8           0          0                no      no      no
1/1/9           0          0                no      no      no
1/1/10          0          0                no      no      no
1/1/11          0          0                no      no      no
1/1/12          0          0                no      no      no
1/1/13          0          0                no      no      no
1/1/14          0          0                no      no      no
1/1/15          0          0                no      no      no
1/1/16          0          0                no      no      no
```

Syntax: show dot1x mac-session brief

The following table describes the information displayed by the **show dot1x mac-session brief** command.

TABLE 38 Output from the **show dot1x mac-session brief** command

Field	Description
Port	Information about the users connected to each port.
Number of users	The number of users connected to the port.
Number of Authorized users	The number of users connected to the port that have been successfully authenticated.
Dynamic VLAN	Whether the port is a member of a RADIUS-specified VLAN.
Dynamic Filters	Whether RADIUS-specified IP ACLs or MAC address filters have been applied to the port.

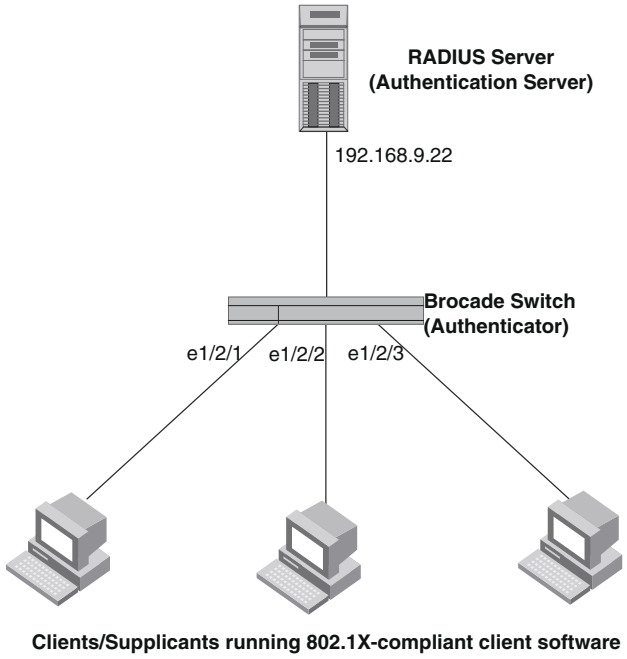
Sample 802.1X configurations

This section illustrates a sample point-to-point configuration and a sample hub configuration that use 802.1X port security.

Point-to-point configuration

Figure 6 illustrates a sample 802.1X configuration with Clients connected to three ports on the Brocade device. In a point-to-point configuration, only one 802.1X Client can be connected to each port.

FIGURE 6 Sample point-to-point 802.1X configuration



Same point-to-point 802.1x configuration

The following commands configure the Brocade device in Figure 6

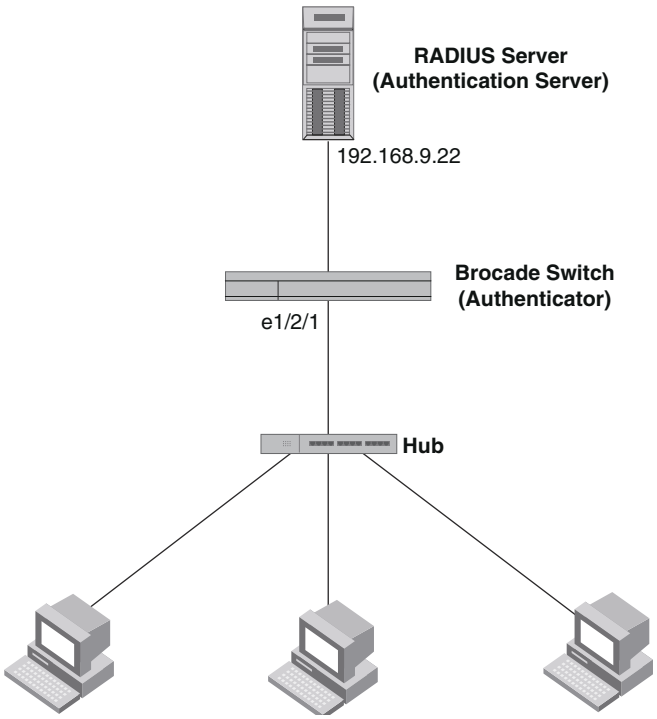
```
Brocade(config)# aaa authentication dot1x default radius
Brocade(config)# radius-server host 192.168.9.22 auth-port 1812 acct-port 1813
default key mirabeau dot1x
Brocade(config)# dot1x-enable ethernet 1/2/1 to 1/2/3
Brocade(config-dot1x)# re-authentication
Brocade(config-dot1x)# timeout re-authperiod 2000
Brocade(config-dot1x)# timeout quiet-period 30
Brocade(config-dot1x)# timeout tx-period 60
Brocade(config-dot1x)# maxreq 6
Brocade(config-dot1x)# exit
Brocade(config)# interface ethernet 1/2/1
Brocade(config-if-e10000-1/2/1)# dot1x port-control auto
Brocade(config-if-e10000-1/2/1)# exit
```

```
Brocade(config)# interface ethernet 1/2/2
Brocadeconfig-if-e10000-1/2/2)# dot1x port-control auto
Brocade(config-if-e10000-1/2/2)# exit
Brocade(config)# interface ethernet 1/2/3
Brocade(config-if-e10000-1/2/3)# dot1x port-control auto
Brocade(config-if-e10000-1/2/3)# exit
```

Hub configuration

Figure 7 illustrates a configuration where three 802.1X-enabled Clients are connected to a hub, which is connected to a port on the Brocade device. The configuration is similar to that in Figure 6, except that 802.1X port security is enabled on only one port, and the **multiple-hosts** command is used to allow multiple Clients on the port.

FIGURE 7 Sample 802.1X configuration using a hub



Sample 802.1x configuration using a hub

The following commands configure the Brocade device in Figure 7

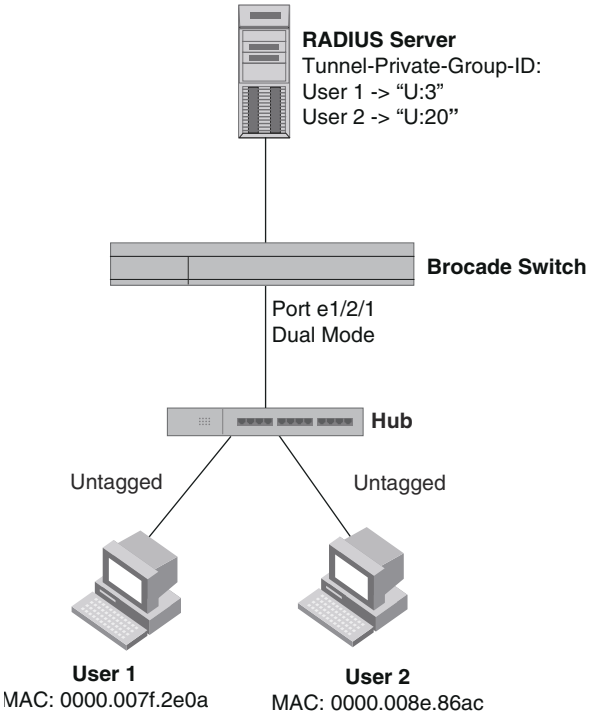
```
Brocade(config)# aaa authentication dot1x default radius
Brocade(config)# radius-server host 192.168.9.22 auth-port 1812 acct-port 1813
default key mirabeau dot1x
Brocade(config)# dot1x-enable ethernet 1/2/1
Brocade(config-dot1x)# re-authentication
Brocade(config-dot1x)# timeout re-authperiod 2000
Brocade(config-dot1x)# timeout quiet-period 30
Brocade(config-dot1x)# timeout tx-period 60
Brocade(config-dot1x)# maxreq 6
Brocade(config-dot1x)# exit
```

```
Brocade(config)#interface ethernet 1/2/1
Brocade(config-if-e10000-1/2/1)# dot1x port-control auto
Brocade(config-if-e10000-1/2/1)# exit
```

802.1X authentication with dynamic VLAN assignment

Figure 8 illustrates 802.1X authentication with dynamic VLAN assignment. In this configuration, two user PCs are connected to a hub, which is connected to port e1/2/1. Port e1/2/1 is configured as a dual-mode port. Both PCs transmit untagged traffic. The profile for User 1 on the RADIUS server specifies that User 1 PC should be dynamically assigned to VLAN 3. The RADIUS profile for User 2 on the RADIUS server specifies that User 2 PC should be dynamically assigned to VLAN 20.

FIGURE 8 Sample configuration using 802.1X authentication with dynamic VLAN assignment



In this example, the PVID for port e1/2/1 would be changed based on the first host to be successfully authenticated. If User 1 is authenticated first, then the PVID for port e1/2/1 is changed to VLAN 3. If User 2 is authenticated first, then the PVID for port e1/2/1 is changed to VLAN 20. Since a PVID cannot be changed by RADIUS authentication after it has been dynamically assigned, if User 2 is authenticated after the port PVID was changed to VLAN 3, then User 2 would not be able to gain access to the network.

If there were only one device connected to the port, and authentication failed for that device, it could be placed into the restricted VLAN, where it could gain access to the network.

The portion of the running-config related to 802.1X authentication is as follows.

```
dot1x-enable
 re-authentication
 servertimeout 10
 timeout re-authperiod 10
 auth-fail-action restricted-vlan
```

```
 auth-fail-vlanid 1023
 mac-session-aging no-aging permitted-mac-only
 enable ethe 1/2/1 to 1/2/4
!
!
!
interface ethernet 1/2/1
 dot1x port-control auto
 dual-mode
```

If User 1 is successfully authenticated before User 2, the PVID for port e1/2/1 would be changed from the default VLAN to VLAN 3.

Had User 2 been the first to be successfully authenticated, the PVID would be changed to 20, and User 1 would not be able to gain access to the network. If there were only one device connected to the port that was sending untagged traffic, and 802.1X authentication failed for that device, it would be placed in the restricted VLAN 1023, and would be able to gain access to the network.

Multi-device port authentication and 802.1X security on the same port

You can configure the Brocade device to use multi-device port authentication and 802.1X security on the same port:

- The multi-device port authentication feature allows you to configure a Brocade device to forward or block traffic from a MAC address based on information received from a RADIUS server. Incoming traffic originating from a given MAC address is switched or forwarded by the device only if the source MAC address is successfully authenticated by a RADIUS server. The MAC address itself is used as the username and password for RADIUS authentication. A connecting user does not need to provide a specific username and password to gain access to the network.
- The IEEE 802.1X standard is a means for authenticating devices attached to LAN ports. Using 802.1X port security, you can configure a Brocade device to grant access to a port based on information supplied by a client to an authentication server.

When both of these features are enabled on the same port, multi-device port authentication is performed prior to 802.1X authentication. If multi-device port authentication is successful, 802.1X authentication may be performed, based on the configuration of a vendor-specific attribute (VSA) in the profile for the MAC address on the RADIUS server.

For more information, including configuration examples, see “Multi-device port authentication and 802.1X security on the same port” on page 234.

Multi-device port authentication and 802.1X security on the same port

Chapter

7

MAC Port Security

Table 39 lists the Media Access Control (MAC) port security features that are supported Brocade ICX 6650. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 39 Supported MAC port security features

Feature	Brocade ICX 6650
MAC port security	Yes
Setting the maximum number of secure MAC addresses on an interface	Yes
Setting the port security age timer	Yes
Specifying secure MAC addresses	Yes
Autosaving secure MAC addresses to the startup-config file	Yes
Specifying the action taken when a security violation occurs	Yes
Clearing port security statistics	Yes

This chapter describes how to configure Brocade devices to learn “secure” MAC addresses on an interface so that the interface will forward only packets that match the secure addresses.

MAC port security overview

You can configure the Brocade device to learn “secure” MAC addresses on an interface. The interface will forward only packets with source MAC addresses that match these learned secure addresses. The secure MAC addresses can be specified manually, or the Brocade device can learn them automatically. After the device reaches the limit for the number of secure MAC addresses it can learn on the interface, if the interface then receives a packet with a source MAC address that does not match the learned addresses, it is considered a security violation.

When a security violation occurs, a syslog entry and an SNMP trap are generated. In addition, the device takes one of two actions: it either drops packets from the violating address (and allows packets from the secure addresses), or disables the port for a specified amount of time. You specify which of these actions takes place.

The secure MAC addresses are flushed when an interface is disabled and re-enabled .

The secure addresses can be kept secure permanently (the default), or can be configured to age out, at which time they are no longer secure. You can configure the device to automatically save the secure MAC address list to the startup-config file at specified intervals, allowing addresses to be kept secure across system restarts.

Local and global resources used for MAC port security

The MAC port security feature uses a concept of local and global “resources” to determine how many MAC addresses can be secured on each interface. In this context, a “resource” is the ability to store one secure MAC address entry. Each interface is allocated 64 local resources. Additional global resources are shared among all interfaces on the device.

When the MAC port security feature is enabled on an interface, the interface can store one secure MAC address. You can increase the number of MAC addresses that can be secured using local resources to a maximum of 64.

Besides the maximum of 64 local resources available to an interface, there are additional global resources. Depending on flash memory size, a device can have 1024, 2048, or 4096 global resources available. When an interface has secured enough MAC addresses to reach its limit for local resources, it can secure additional MAC addresses by using global resources. Global resources are shared among all the interfaces on a first-come, first-served basis.

The maximum number of MAC addresses any single interface can secure is 64 (the maximum number of local resources available to the interface), plus the number of global resources not allocated to other interfaces.

Configuration notes and feature limitations for MAC port security

The following limitations apply to this feature:

- MAC port security applies only to Ethernet interfaces.
- MAC port security is not supported on static trunk group members or ports that are configured for link aggregation.
- MAC port security is not supported on 802.1X port security-enabled ports.

- Brocade devices do not support the **reserved-vlan-id** *num* command, which changes the default VLAN ID for the MAC port security feature.
- The SNMP trap generated for restricted MAC addresses indicates the VLAN ID associated with the MAC address, as well as the port number and MAC address.
- MAC port security is not supported on ports that have multi-device port authentication enabled.
- The first packet from each new secure MAC address is dropped if secure MAC addresses are learned dynamically.

MAC port security configuration

To configure the MAC port security feature, perform the following tasks:

- Enable the MAC port security feature
- Set the maximum number of secure MAC addresses for an interface
- Set the port security age timer
- Specify secure MAC addresses
- Configure the device to automatically save secure MAC addresses to the startup-config file
- Specify the action taken when a security violation occurs

Enabling the MAC port security feature

By default, the MAC port security feature is disabled on all interfaces. You can enable or disable the feature on all interfaces at once, or on individual interfaces.

To enable the feature on all interfaces at once, enter the following commands.

```
Brocade(config)# port security
Brocade(config-port-security)# enable
```

To disable the feature on all interfaces at once, enter the following commands.

```
Brocade(config)# port security
Brocade(config-port-security)# no enable
```

To enable the feature on a specific interface, enter the following commands.

```
Brocade(config)# interface ethernet 1/1/7
Brocade(config-if-e10000-1/1/7)# port security
Brocade(config-port-security-e10000-1/1/7)# enable
```

Syntax: port security

Syntax: [no] enable

Setting the maximum number of secure MAC addresses for an interface

When MAC port security is enabled, an interface can store one secure MAC address. You can increase the number of MAC addresses that can be stored to a maximum of 64, plus the total number of global resources available.

For example, to configure interface 1/1/7 to have a maximum of 10 secure MAC addresses, enter the following commands.

```
Brocade(config)# interface ethernet 1/1/7
Brocade(config-if-e10000-1/1/7)# port security
Brocade(config-port-security-e10000-1/1/7)# maximum 10
```

Syntax: maximum *number-of-addresses*

The *number-of-addresses* parameter can be set to a number from 0 through 64 plus (the total number of global resources available). The total number of global resources is 2048 or 4096, depending on flash memory size. Setting the parameter to 0 prevents any addresses from being learned. The default is 1.

Setting the port security age timer

By default, learned MAC addresses stay secure indefinitely. You can optionally configure the device to age out secure MAC addresses after a specified amount of time.

To set the port security age timer to 10 minutes on all interfaces, enter the following commands.

```
Brocade(config)# port security
Brocade(config-port-security)# age 10
```

To set the port security age timer to 10 minutes on a specific interface, enter the following commands.

```
Brocade(config)# interface ethernet 1/1/7
Brocade(config-if-e10000-1/1/7)# port security
Brocade(config-port-security-e10000-1/1/7)# age 10
```

Syntax: [**no**] **age** *minutes*

The *minutes* variable specifies a range from 0 through 1440 minutes.The default is 0 (never age out secure MAC addresses).

NOTE
Even though you can set age time to specific ports independent of the device-level setting, the actual age timer will take the greater of the two values. Thus, if you set the age timer to 3 minutes for the port, and 10 minutes for the device, the port MAC aging happens in 10 minutes (the device-level setting), which is greater than the port setting that you have configured.

Specifying secure MAC addresses

You can configure secure MAC addresses on tagged and untagged interfaces.

On an untagged interface

To specify a secure MAC address on an untagged interface, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/1/7
Brocade(config-if-e10000-1/1/7)# port security
Brocade(config-port-security-e10000-1/1/7)# secure-mac-address 0000.0018.747C
```

Syntax: [**no**] **secure-mac-address** *mac-address*

On a tagged interface

When specifying a secure MAC address on a tagged interface, you must also specify the VLAN ID. To do so, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/1/7
Brocade(config-if-e10000-1/1/7)# port security
Brocade(config-port-security-e10000-1/1/7)# secure-mac-address 0000.0018.747C 2
```

Syntax: [**no**] **secure-mac-address** *mac-address vlan-ID*

NOTE
If MAC port security is enabled on a port and you change the VLAN membership of the port, make sure that you also change the VLAN ID specified in the **secure-mac-address** configuration statement for the port.

When a secure MAC address is applied to a tagged port, the VLAN ID is generated for both tagged and untagged ports. When you display the configuration, you will see an entry for the secure MAC addresses. For example, you might see an entry similar to the following line.

```
secure-mac-address 0000.0011.2222 10
```

This line means that MAC address 0000.0011.2222 on VLAN 10 is a secure MAC address.

Autosaving secure MAC addresses to the startup configuration

Learned MAC addresses can automatically be saved to the startup configuration at specified intervals. The autosave feature saves learned MAC addresses by copying the running configuration to the startup configuration.

For example, to automatically save learned secure MAC addresses every 20 minutes, enter the following commands.

```
Brocade(config)# port security
Brocade(config-port-security)# autosave 20
```

Syntax: [**no**] **autosave** *minutes*

The *minutes* variable can be from 15 through 1440 minutes. By default, secure MAC addresses are not autosaved to the startup-config file.

If you change the autosave interval, the next save happens according to the old interval, then the new interval takes effect. To change the interval immediately, disable autosave by entering the **no autosave** command, then configure the new autosave interval using the **autosave** command.

Specifying the action taken when a security violation occurs

A security violation can occur when a user tries to connect to a port where a MAC address is already locked, or the maximum number of secure MAC addresses has been exceeded. When a security violation occurs, an SNMP trap and syslog message are generated.

You can configure the device to take one of two actions when a security violation occurs; either drop packets from the violating address (and allow packets from secure addresses), or disable the port for a specified time.

Dropping packets from a violating address

To configure the device to drop packets from a violating address and allow packets from secure addresses, enter the following commands.

```
Brocade(config)# interface ethernet 1/1/7
Brocade(config-if-e10000-1/1/7)# port security
Brocade(config-port-security-e10000-1/1/7)# violation restrict
```

Syntax: violation [restrict]

NOTE
When the **restrict** option is used, the maximum number of MAC addresses that can be restricted is 128. If the number of violating MAC addresses exceeds this number, the port is shut down. An SNMP trap and the following Syslog message are generated: "Port Security violation restrict limit 128 exceeded on interface ethernet *port_id*". This is followed by a port shutdown Syslog message and trap.

Specifying the period of time to drop packets from a violating address

To specify the number of minutes that the device drops packets from a violating address, use commands similar to the following.

```
Brocade(config)# interface ethernet 1/1/7
Brocade(config-if-e10000-1/1/7)# port security
Brocade(config-port-security-e10000-1/1/7)# violation restrict 5
```

Syntax: violation restrict *age*

The *age* variable can be from 0 through 1440 minutes. The default is 5 minutes. Specifying 0 drops packets from the violating address permanently.

Aging for restricted MAC addresses is done in software. There can be a worst case inaccuracy of one minute from the specified time.

The restricted MAC addresses are denied in hardware.

Disabling the port for a specified amount of time

You can configure the device to disable the port for a specified amount of time when a security violation occurs.

To shut down the port for 5 minutes when a security violation occurs, enter the following commands.

```
Brocade(config)# interface ethernet 1/1/7
Brocade(config-if-e10000-1/1/7)# port security
Brocade(config-port-security-e10000-1/1/7)# violation shutdown 5
```

Syntax: violation shutdown *minutes*

The minutes can be from 0 through 1440 minutes. Specifying 0 shuts down the port permanently when a security violation occurs.

Clearing port security statistics

You can clear restricted MAC addresses and violation statistics from ports on all ports or on individual ports.

Clearing restricted MAC addresses

To clear all restricted MAC addresses globally, enter the **clear port security restricted-macs all** command.

```
Brocade# clear port security restricted-macs all
```

To clear restricted MAC addresses on a specific port, enter a command such as the following.

```
Brocade# clear port security restricted-macs ethernet 1/1/5
```

Syntax: clear port security restricted-macs all | **ethernet** *port*

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

Clearing violation statistics

To clear violation statistics globally, enter the **clear port security statistics** all command.

```
Brocade# clear port security statistics all
```

To clear violation statistics on a specific port, enter a command such as the following.

```
Brocade# clear port security statistics ethernet 1/1/5
```

Syntax: clear port security statistics all | **ethernet** *port*

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

Displaying port security information

You can display the following information about the MAC port security feature:

- The port security settings for an individual port or for all the ports on a specified module
- The secure MAC addresses configured on the device
- Port security statistics for an interface or for a module

Displaying port security settings

You can display the port security settings for an individual port or for all the ports on a specified module. For example, to display the port security settings for port 7/11, enter the following command.

```
Brocade# show port security ethernet 1/1/7
Port   Security Violation Shutdown-Time Age-Time  Max-MAC
------ -------- --------- ------------- --------- -------
1/1/7 disabled  shutdown            10        10       1
```

Syntax: show port security ethernet *port*

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

TABLE 40 Output from the **show port security ethernet** command

Field	Description
Port	The slot and port number of the interface.
Security	Whether the port security feature has been enabled on the interface.
Violation	The action to be undertaken when a security violation occurs, either “shutdown” or “restrict”.
Shutdown-Time	The number of seconds a port is shut down following a security violation, if the port is set to “shutdown” when a violation occurs.
Age-Time	The amount of time, in minutes, MAC addresses learned on the port will remain secure.
Max-MAC	The maximum number of secure MAC addresses that can be learned on the interface.

Displaying the secure MAC addresses

To list the secure MAC addresses configured on the device, enter the following command.

```
Brocade# show port security mac
Port   Num-Addr Secure-Src-Addr Resource Age-Left  Shutdown/Time-Left
------ -------- --------------- -------- --------- ------------------
1/1/7     1     0000.0018.747c   Local        10       no
```

Syntax: show port security mac

Table 41 describes the output from the **show port security mac** command.

TABLE 41 Output from the **show port security mac** command

Field	Description
Port	The slot and port number of the interface.
Num-Addr	The number of MAC addresses secured on this interface.
Secure-Src-Addr	The secure MAC address.
Resource	Whether the address was secured using a local or global resource.Refer to "Local and global resources used for MAC port security" on page 202 for more information.
Age-Left	The number of minutes the MAC address will remain secure.
Shutdown/Time-Left	Whether the interface has been shut down due to a security violation and the number of seconds before it is enabled again.

NOTE
After every switchover or failover, the MAC "Age-Left" timer is reset to start since it is not synchronized between the master and the standby stack unit.

Displaying port security statistics

You can display port security statistics for an interface or for a module.

For example, to display port security statistics for interface 1/1/7, enter the following command.

```
Brocade# show port security statistics e 1/1/7
Port  Total-Addrs Maximum-Addrs Violation Shutdown/Time-Left
----- ----------- ------------- --------- ------------------
1/1/7      1             1          0        no
```

Syntax: show port security statistics *port*

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

TABLE 42 Output from the **show port security statistics** ***port*** command

Field	Description
Port	The slot and port number of the interface.
Total-Addrs	The total number of secure MAC addresses on the interface.
Maximum-Addrs	The maximum number of secure MAC addresses on the interface.
Violation	The number of security violations on the port.
Shutdown/Time-Left	Whether the port has been shut down due to a security violation and the number of seconds before it is enabled again.

For example, to display port security statistics for interface module 7, enter the **show port security statistics** command.

```
Brocade# show port security statistics 7
Module 7:
  Total ports: 0
  Total MAC address(es): 0
  Total violations: 0
  Total shutdown ports 0
```

Syntax: show port security statistics *module*

Table 43 describes the output from the **show port security statistics** *module* command.

TABLE 43 Output from the **show port security statistics *module*** command

Field	Description
Total ports	The number of ports on the module.
Total MAC address(es)	The total number of secure MAC addresses on the module.
Total violations	The number of security violations encountered on the module.
Total shutdown ports	The number of times that ports on the module shut down as a result of security violations.

Displaying restricted MAC addresses on a port

To display a list of restricted MAC addresses on a port, enter a command such as the following.

```
Brocade# show port security ethernet 1/1/5 restricted-macs
```

Syntax: show port security ethernet *port* **restricted-macs**

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

Chapter

MAC-based VLANs

8

Table 44 lists the MAC-based VLAN features that are supported on Brocade ICX 6650 device. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 44 Supported MAC-based VLAN features

Feature	Brocade ICX 6650
MAC-Based VLANs: • Source MAC address authentication • Policy-based classification and forwarding	Yes
MAC-based VLANs and 802.1X security on the same port	Yes
MAC-based VLAN aging	Yes
Dynamic MAC-Based VLANs	Yes

MAC-based VLAN overview

The MAC-based VLAN feature controls network access by authenticating a host source MAC address, and mapping the incoming packet source MAC to a VLAN. Mapping is based on the MAC address of the end station connected to the physical port. Users who relocate can remain on the same VLAN as long as they connect to any switch in the same domain, on a port which is permitted in the VLAN. The MAC-based VLAN feature may be enabled for two types of hosts: static and dynamic.

MAC-based VLAN activity is determined by authentication through a RADIUS server. Incoming traffic that originates from a specific MAC address is forwarded only if the source MAC address-to-VLAN mapping is successfully authenticated. While multi-device port authentication is in progress, all traffic from the new MAC address will be blocked or dropped until the authentication succeeds. Traffic is dropped if the authentication fails.

Static and dynamic hosts

Static hosts are devices on the network that do not speak until spoken to. Static hosts may not initiate a request for authentication on their own. Such static hosts can be managed through a **link up** or **link down** notification.

Dynamic hosts are “chatty” devices that generate packets whenever they are in the **link up** state. Dynamic hosts must be authenticated before they can switch or forward traffic.

MAC-based VLAN feature structure

The MAC-based VLAN feature operates in two stages:

- Source MAC Address Authentication
- Policy-Based Classification and Forwarding

Source MAC address authentication

Source MAC address authentication is performed by a central RADIUS server when it receives a PAP request with a username and password that match the MAC address being authenticated. When the MAC address is successfully authenticated, the server must return the VLAN identifier, which is carried in the Tunnel-Type, Tunnel-Medium-Type, and Tunnel-Private-Group-ID attributes of the RADIUS packets. If the Tunnel-Type is *tagged*, the MAC address will be blocked or restricted. If the identified VLAN does not exist, then the authentication is considered a failure, and action is taken based on the configured failure options. (The default failure action is to drop the traffic.) The RADIUS server may also optionally return the QoS attribute for the authenticated MAC address. Refer to Table 47 on page 217 for more information about attributes.

Policy-based classification and forwarding

After the authentication stage is complete, incoming traffic is classified based on the response from the RADIUS server. There are three possible actions:

- Incoming traffic from a specific source MAC is dropped because authentication failed
- Incoming traffic from a specific source MAC is classified as untagged into a specific VLAN
- Incoming traffic from a specific source MAC is classified as untagged into a restricted VLAN

Traffic classification is performed by programming incoming traffic and RADIUS-returned attributes in the hardware. Incoming traffic attributes include the source MAC address and the port on which the feature is enabled. The RADIUS-returned attributes are the VLAN into which the traffic is to be classified, and the QoS priority.

NOTE
This feature drops any incoming *tagged* traffic on the port, and classifies and forwards untagged traffic into the appropriate VLANs.

This feature supports up to a maximum of 32 MAC addresses per physical port, with a default of 2.

NOTE
Even though the feature supports up tp a maximum of 32 MAC address per physical port, the configuration of the maximum number of MAC addresses per port is limited by the available hardware resources.

Once a client MAC address is successfully authenticated and registered, the MAC-to-VLAN association remains until the port connection is dropped, or the MAC entry expires.

MAC-based VLAN and port up or down events

When the state of a port is changed to *down*, all authorized and unauthorized MAC addresses are removed from the MAC-to-VLAN mapping table, any pending authentication requests are cancelled.

Dynamic MAC-based VLAN

When enabled, the dynamic MAC-based VLAN feature allows the dynamic addition of mac-vlan-permit ports to the VLAN table only after successful RADIUS authentication. Ports that fail RADIUS authentication are not added to the VLAN table.

When this feature is not enabled, the physical port is statically added to the hardware table, regardless of the outcome of the authentication process. This feature prevents the addition of unauthenticated ports to the VLAN table. For information about how to configure Dynamic MAC-based VLAN, refer to "Configuring dynamic MAC-based VLAN" on page 220.

Configuration notes and feature limitations for dynamic MAC-based VLAN

The following guidelines apply to MAC-based VLAN configurations:

- MAC-based VLAN is not currently supported for trunk ports and LACP.
- MAC-based VLAN is not supported for VLAN groups, topology groups and dual-mode configuration.
- MAC-based VLAN is not supported together with ACLs or MAC address filters.
- Brocade ICX 6650 devices do not support UDLD link-keepalives on ports with MAC-based VLAN enabled.
- Brocade ICX 6650 devices do not support STP BPDU packets on ports with MAC-based VLAN enabled.
- MAC-to-VLAN mapping must be associated with VLANs that exist on the switch. Create the VLANs before you configure the MAC-based VLAN feature.
- Ports participating in MAC-based VLANs must first be configured as **mac-vlan-permit** ports under the VLAN configuration.
- In the RADIUS server configuration file, a MAC address cannot be configured to associate with more than one VLAN.
- This feature does not currently support dynamic assignment of a port to a VLAN. Users must pre-configure VLANs and port membership before enabling the feature.
- Multi-device port authentication filters will not work with MAC-based VLANs on the same port.

Dynamic MAC-based VLAN CLI commands

The following table describes the CLI commands used to configure MAC-based VLANs.

TABLE 45 CLI commands for MAC-based VLANs

CLI command	Description	CLI level
mac-auth mac-vlan enable	Enables per-port MAC-based VLAN	Interface
mac-auth mac-vlan disable	Disables per-port MAC-based VLAN	interface
mac-auth mac-vlan-dyn-activation	Enables Dynamic MAC-based VLAN	global
no mac-auth mac-vlan-dyn-activation	Disables Dynamic MAC-based VLAN	global
no mac-auth mac-vlan	Removes the MAC-VLAN configuration from the port	interface

TABLE 45 CLI commands for MAC-based VLANs (Continued)

CLI command	Description	CLI level
mac-auth mac-vlan max-mac-entries *num-of-entries*	The maximum number of allowed and denied MAC addresses (static and dynamic) that can be learned on a port. The default is 2.	interface
mac-auth mac-vlan *mac-addr* vlan *vlan-id* priority <*0-7*>	Adds a static MAC-VLAN mapping to the MAC-based VLAN table (for static hosts)	interface
clear table-mac-vlan	Clears the contents of the authenticated MAC address table	global
clear table-mac-vlan ethernet *port*	Clears all MAC-based VLAN mapping on a port	global
show table-mac-vlan	Displays information about allowed and denied MAC addresses on ports with MAC-based VLAN enabled.	global
show table-mac-vlan allowed-mac	Displays MAC addresses that have been successfully authenticated	global
show table-mac-vlan denied-mac	Displays MAC addresses for which authentication failed	global
show table-mac-vlan detailed	Displays detailed MAC-VLAN settings and classified MAC addresses for a port with the feature enabled	global
show table-mac-vlan *mac-address*	Displays status and details for a specific MAC address	global
show table-mac-vlan ethernet *port*	Displays all MAC addresses allowed or denied on a specific port	global

Dynamic MAC-based VLAN configuration example

The following example shows a MAC-based VLAN configuration.

```
Brocade# show run
Current configuration:
ver 04.0.00b122T7e1
fan-threshold mp speed-3 35 100
module 1 icx6650-64-56-port-management-module
module 2 icx6650-64-4-port-160g-module
module 3 icx6650-64-8-port-80g-module
vlan 1 by port
 untagged ethernet 1/1/10
 mac-vlan-permit ethernet 1/1/1 to 1/1/3
 no spanning-tree
vlan 2 by port
 untagged ethernet 1/1/24
 mac-vlan-permit ethernet 1/1/1 to 1/1/3
 no spanning-tree
vlan 222 name RESTRICTED_MBV by port
 untagged ethe 1/1/4
 mac-vlan-permit ethernet 1/1/1 to 1/1/3
vlan 666 name RESTRICTED_MAC_AUTH by port
 untagged ethe 1/1/20
 mac-vlan-permit ethernet 1/1/1 to 1/1/3
 spanning-tree 802-1w
vlan 4000 name DEFAULT-VLAN by port
```

```
vlan 4004 by port
 mac-vlan-permit ethernet 1/1/1 to 1/1/3
default-vlan-id 4000
ip address 10.44.3.3 255.255.255.0
ip default-gateway 10.44.3.1
radius-server host 10.44.3.111
radius-server key 1 $-ndUno
mac-authentication enable
mac-authentication mac-vlan-dyn-activation
mac-authentication max-age 60
mac-authentication hw-deny-age 30
mac-authentication auth-passwd-format xxxx.xxxx.xxxx
mac-authentication auth-fail-vlan-id 666
interface ethernet 1/1/1
 mac-authentication mac-vlan max-mac-entries 5
 mac-authentication mac-vlan 0030.4888.b9fe vlan 1 priority 1
 mac-authentication mac-vlan enable
interface ethernet 1/1/2
 mac-authentication mac-vlan max-mac-entries 10
 mac-authentication mac-vlan enable
 mac-authentication auth-fail-action restrict-vlan 222
interface ethernet 1/1/3
 mac-authentication mac-vlan enable
 mac-authentication auth-fail-action restrict-vlan
!
end
```

MAC-based VLAN configuration

Configure MAC-based VLAN mapping on the switch statically for static hosts, or dynamically for non-static hosts, by directing the RADIUS server to authenticate the incoming packet.

To configure the a MAC-based VLAN, first perform the following tasks:

- In the VLANs, configure **mac-vlan-permit** for each port that will be participating in the MAC-based VLAN
- If a port has been MAC-based VLAN-enabled, but has **not** been added as **mac-vlan-permit** in any of the VLANs, any MAC addresses learned on this port will be blocked in the reserved VLAN. To prevent this, you must create all of the VLANs and add all ports as **mac-vlan-permit** ***before*** enabling MAC-based VLAN on any ports.
- Disable any multi-device port authentication on ports you will be using for MAC-to-VLAN mapping

NOTE
Do not configure MAC-based VLAN on ports that are tagged to any VLAN. Do not use ports on which MAC-based VLAN is configured as tagged ports.

NOTE
MAC-based VLAN is not supported on trunk or LACP ports. Do not configure trunks on MAC-based VLAN-enabled ports.

Using MAC-based VLANs and 802.1X security on the same port

On Brocade devices, MAC-based VLANs and 802.1X security can be configured on the same port. When both of these features are enabled on the same port, MAC-based VLAN is performed prior to 802.1X authentication. If MAC-based VLAN is successful, 802.1X authentication may be performed, based on the configuration of a vendor-specific attribute (VSA) in the profile for the MAC address on the RADIUS server.

When both features are configured on a port, a device connected to the port is authenticated as follows.

1. MAC-based VLAN is performed on the device to authenticate the device MAC address.
2. If MAC-based VLAN is successful, the device then checks to see if the RADIUS server included the Foundry-802_1x-enable VSA (described in Table 47) in the Access-Accept message that authenticated the device.
3. If the Foundry-802_1x-enable VSA is not present in the Access-Accept message, or is present and set to 1, then 802.1X authentication is performed for the device.
4. If the Foundry-802_1x-enable VSA is present in the Access-Accept message, and is set to 0, then 802.1X authentication is skipped.

Configuring generic and Brocade vendor-specific attributes on the RADIUS server

If the RADIUS authentication process is successful, the RADIUS server sends an Access-Accept message to the Brocade device, authenticating the device. The Access-Accept message includes Vendor-Specific Attributes (VSAs) that specify additional information about the device.

Add Brocade vendor-specific attributes to your RADIUS server configuration, and configure the attributes in the individual or group profiles of the devices that will be authenticated. Brocade. vendor-ID is 1991, vendor-type 1. Table 46 lists generic RADIUS attributes. Table 47 lists Brocade Vendor-Specific Attributes.

TABLE 46 Generic RADIUS attributes

Attribute name	Attribute ID	Data type	Optional or mandatory	Description
Tunnel-Type	64	13 decimal VLAN	Mandatory	RFC 2868.
Tunnel-Medium-Type	65	6 decimal 802	Mandatory	RFC 2868.
Tunnel-Private-Group-ID	81	decimal	Mandatory	RFC 2868. *vlan-id* or U:*vlan -id* – a MAC-based VLAN ID configured on the Brocade device.

TABLE 47 Brocade vendor-specific attributes for RADIUS

Attribute name	Attribute ID	Data type	Optional or mandatory	Description
Foundry-MAC-based VLAN-QoS	8	decimal	Optional	The QoS attribute specifies the priority of the incoming traffic based on any value between 0 (lowest priority) and 7 (highest priority). Default is 0.
Foundry-802_1x-enable	6	integer	Optional	Specifies whether 802.1X authentication is performed when MAC-based VLAN is successful for a device. This attribute can be set to one of the following: 0 - Do not perform 802.1X authentication on a device that passes MAC-based VLAN. Set the attribute to zero (0) for devices that do not support 802.1X authentication. 1 - Perform 802.1X authentication when a device passes MAC-based VLAN. Set the attribute to one (1) for devices that support 802.1X authentication.
Foundry-802_1x-valid	7	integer	Optional	Specifies whether the RADIUS record is valid only for MAC-based VLAN, or for both MAC-based VLAN and 802.1X authentication. This attribute can be set to one of the following: 0 - The RADIUS record is valid only for MAC-based VLAN. Set this attribute to zero (0) to prevent a user from using their MAC address as username and password for 802.1X authentication 1 - The RADIUS record is valid for both MAC-based VLAN and 802.1X authentication.

Aging for MAC-based VLAN

The aging process for MAC-based VLAN works as described below.

For permitted hosts

For permitted hosts, as long as the Brocade device is receiving traffic aging does not occur. The age column in the output of the **show table-mac-vlan** command displays Ena or S *num*. If the Brocade device stops receiving traffic, the entry first ages out from the MAC table (in the hardware) and then the aging cycle for MAC-based VLAN begins. Aging in the MAC-based VLAN continues for 2 minutes (the default is 120 seconds) after which the MAC-based VLAN session is flushed out.

For blocked hosts

For blocked hosts, as long as the Brocade device is receiving traffic, aging does not occur. In the output of the **show table-mac-vlan command**, the age column displays H0 to H70, S0, and H0 to H70, etc. Aging of the MAC-based VLAN MAC occurs in two phases: hardware aging and software aging. The hardware aging period can be configured using the **mac-authentication hw-deny-age** command in config mode. The default is 70 seconds. The software aging time for MAC-based VLAN MACs can be configured using the **mac-authentication max-age** command. When the Brocade device is no longer receiving traffic from a MAC-based VLAN MAC address, the hardware aging

period begins and lasts for a fixed length of time (default or user-configured). When the hardware aging period ends, the software aging period begins. The software aging period lasts for a configurable amount of time (the default is 120 seconds). After the software aging period ends, the MAC-based VLAN session is flushed, and the MAC address can be authenticated or denied if the Brocade device again receives traffic from that MAC address.

For MAC-based dynamic activation

If all of the sessions age out on a port, the port is dynamically removed from the VLAN table. When any new session is established, the port is dynamically added back to the VLAN table.

NOTE
If the Brocade device receives a packet from an authenticated MAC address, and the MAC-based VLAN software aging is still in progress (hardware aging has already occurred), a RADIUS message is NOT sent to the RADIUS server. Instead the MAC address is reentered in the hardware along with the parameters previously returned from the RADIUS server. A RADIUS message is sent only when the MAC-based VLAN session ages out from the software.

To change the length of the software aging period

To change the length of the software aging period for blocked MAC addresses, enter a command such as the following.

```
Brocade(config)# mac-authentication max-age 180
```

Syntax: [**no**] **mac-authentication max-age** *seconds*

You can specify from 1–65535 seconds. The default is 120 seconds.

Disabling aging for MAC-based VLAN sessions

MAC addresses that have been authenticated or denied by a RADIUS server are aged out if no traffic is received from the MAC address for a certain period of time.

You can optionally disable aging for MAC-based VLAN session subject to authentication, either for all MAC addresses or for those learned on a specified interface.

Globally disabling aging

On most devices, you can disable aging on all interfaces where MAC-based VLAN has been enabled, by entering the following command.

```
Brocade(config)# mac-authentication disable-aging
```

Syntax: mac-authentication disable-aging

Enter the command at the global or interface configuration level.

The **denied-mac-only** parameter prevents denied sessions from being aged out, but ages out permitted sessions.

The **permitted-mac-only** parameter prevents permitted (authenticated and restricted) sessions from being aged out and ages denied sessions.

Disabling the aging on interfaces

To disable aging on a specific interface where MAC-based VLAN has been enabled, enter the command at the interface level.

```
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# mac-authentication disable-aging
```

Syntax: [no] mac-authentication disable-aging

Configuring the maximum MAC addresses per port

To configure the maximum number of MAC addresses allowed per port, use the following commands:

```
Brocade(config)# interface ethernet 1/1/1
Brocade(config-if-e10000-1/1/1)# mac-authentication mac-vlan max-mac-entries 24
```

NOTE
32 MAC addresses maximum are allowed per port. This total includes both static and dynamic hosts. The default number of allowed MACs is 2. Even though the feature supports up tp a maximum of 32 MAC address per physical port, the configuration of the maximum number of MAC addresses per port is limited by the available hardware resources.

NOTE
To change the maximum MAC addresses per port, you must first disable MAC-based VLAN on that port.

Configuring a MAC-based VLAN for a static host

Follow the steps given below to configure a MAC-based VLAN for a static host.

1. Enable multi-device port authentication globally using the following command.

   ```
   Brocade(config)# mac-authentication enable
   ```

2. Add each port on which you want MAC-based VLAN enabled as mac-vlan-permit for a specific VLAN.

   ```
   Brocade(config)# vlan 10 by port
   Brocade(config-vlan-10)# mac-vlan-permit ethernet 1/1/1 to 1/1/6
   added mac-vlan-permit ports ethe 1/1/1 to 1/1/6 to port-vlan 10.
   ```

3. Add the static MAC-based VLAN configuration on the port.

   ```
   Brocade(config)# interface ethernet 1/1/1
   Brocade(config-if-e10000-1/1/1)# mac-authentication mac-vlan 0000.0010.0011
   vlan 10 priority 5
   ```

4. To enable MAC-based VLAN on the port.

   ```
   Brocade(config)# interface ethernet 1/1/1
   Brocade(config-if-e10000-1/1/1)# mac-authentication mac-vlan enable
   ```

5. To disable MAC-based VLAN on the port.

   ```
   Brocade(config)# interface ethernet 1/1/1
   Brocade(config-if-e10000-1/1/1)# mac-auth mac-vlan disable
   ```

6. To remove and disable the MAC-based VLAN configuration.

```
Brocade(config)# interface ethernet 1/1/1
Brocade(config-if-e10000-1/1/1)# no mac-auth mac-vlan
```

Configuring MAC-based VLAN for a dynamic host

Follow the steps given below to configure MAC-based VLAN for a dynamic host.

1. Enable multi-device port authentication globally using the following command.

```
Brocade(config)# mac-authentication enable
```

2. Add each port on which you want MAC-based VLAN enabled as **mac-vlan-permit** for a specific VLAN.

```
Brocade(config)# vlan 10 by port
Brocade(config-vlan-10)# mac-vlan-permit ethernet 1/1/1 to 1/1/6
```

3. Enable MAC-based VLAN on the port.

```
Brocade(config)# interface ethernet 1/1/1
Brocade(config-if-e10000-1/1/1)# mac-authentication mac-vlan enable
```

4. Disable MAC-based VLAN on the port.

```
Brocade(config)# interface ethernet 1/1/1
Brocade(config-if-e10000-1/1/1)# mac-auth mac-vlan disable
```

5. Remove and disable the MAC-based VLAN configuration.

```
Brocade(config)# interface ethernet 1/1/1
Brocade(config-if-e10000-1/1/1)# no mac-auth mac-vlan
```

Configuring dynamic MAC-based VLAN

To globally enable MAC-based VLAN globally (for all MAC-based VLAN ports), enter the following commands.

```
Brocade(config)# mac-authentication enable
Brocade(config)# mac-authentication mac-vlan-dyn-activation
```

To configure Dynamic MAC-based VLAN to add a specific port to a specific VLAN, enter commands similar to the following.

```
Brocade(config)# vlan 10
Brocade(config-vlan-10)# mac-vlan-permit ethernet 1/1/5
```

Syntax: mac-vlan-permit ethernet *stack-unit/slotnum/portnum*

To disable Dynamic MAC-based VLAN, enter the following command.

```
Brocade(config)# no mac-authentication mac-vlan-dyn-activation
```

NOTE
If static Mac-Based VLAN is configured on a port, the port will be added only to the VLAN table for which the static MAC-based VLAN configuration exists.

NOTE
If the Dynamic MAC-based VLAN is enabled after any MAC-based VLAN sessions are established, all sessions are flushed and the mac-vlan-permit ports are removed from the VLAN. The ports are then added back to the VLAN dynamically after they successfully pass the RADIUS authentication process.

Configuring MAC-based VLANs using SNMP

Several MIB objects have been developed to allow the configuration of MAC-based VLANs using SNMP. For more information, refer to the *IronWare MIB Reference Guide*.

Displaying information about MAC-based VLANs

This section describes the **show** commands that display information related to MAC-based VLANs.

Displaying the MAC-VLAN table

Enter the following command to display the MAC-VLAN table.

```
Brocade(config)# show table-mac-vlan
-----------------------------------------------------------------
Port    Vlan  Accepted  Rejected  Attempted  Static  Static  Max
              Macs      Macs      Macs       Macs    Conf    Macs
-----------------------------------------------------------------
1/1/1   N/A   1         1         0          0       1       10
```

Syntax: show table-mac-vlan

The following table describes the information in this output.

TABLE 48 Output description of **show table-mac-vlan** command

Field	Description
Port	The port number where MAC-based VLAN is enabled.
Vlan	Not applicable for this feature, will always display n/a.
Accepted Macs	The number of MAC addresses that have been successfully authenticated (dynamic hosts) combined with the number of active static MAC addresses (static hosts).
Rejected Macs	The number of MAC addresses for which authentication has failed for dynamic hosts.
Attempted Macs	The number of attempts made to authenticate MAC addresses.
Static Macs	The number of currently connected active static hosts.
Static Conf	The number of static hosts that are configured on the physical port.
Max Macs	The maximum number of allowed MAC addresses.

Displaying the MAC-VLAN table for a specific MAC address

Enter the **show table-mac-vlan** command to display the MAC-VLAN table information for a specific MAC address.

```
Brocade(config)# show table-mac-vlan 0000.0010.1001
------------------------------------------------------------------------------
MAC Address     Port       Vlan Authenticated Time    Age   dot1x
------------------------------------------------------------------------------
0000.0010.1001   1/1/1       2    Yes    00d00h05m45s Ena    Dis
```

Syntax: show table-mac-vlan *mac-address*

The following table describes the information in this output.

TABLE 49 Output field description of **show table-mac-vlan** command

Field	Description
MAC Address	The MAC address for which this information is displayed.
Port	The port where MAC-based VLAN is enabled.
Vlan	The VLAN to which the MAC address has been assigned.
Authenticated	Yes indicates authentication is successful. No indicates authentication has failed. Inp indicates authentication in progress Rst indicates a restricted VLAN
Time	The time at which the MAC address was authenticated. If the clock is set on the Brocade device, then the actual date and time are displayed. If the clock has not been set, then the time is displayed relative to when the device was last restarted.
Age	The age of the MAC address entry in the authenticated MAC address list.
Dot1x	Indicates if 802.1X authentication is enabled or disabled for the MAC address.

Displaying allowed MAC addresses

Enter the **show table-mac-vlan allowed-mac** command to display information about successfully authenticated MAC addresses.

```
Brocade# show table-mac-vlan allowed-mac
------------------------------------------------------------------------------
MAC Address     Port       Vlan Authenticated Time    Age   dot1x
------------------------------------------------------------------------------
0000.0074.3181   1/1/17      76   Yes    00d01h17m22s Ena    Dis
```

Syntax: show table-mac-vlan allowed-mac

The following table describes the information in this output.

TABLE 50 Output field description of the **show table-mac-vlan allowed-mac** command

Field	Description
MAC Address	The allowed MAC addresses for which the information is displayed.
Port	The port where MAC-based VLAN is enabled.

TABLE 50 Output field description of the **show table-mac-vlan allowed-mac** command

Field	Description
Vlan	The VLAN to which the MAC address has been assigned.
Authenticated	Yes indicates authentication has been successful. Inp indicates authentication is in progress.
Time	The time at which each MAC address was authenticated. If the clock is set on the Brocade device, then the actual date and time are displayed. If the clock has not been set, then the time is displayed relative to when the device was last restarted.
Age	The age of the MAC address entry in the authenticated MAC address list.
Dot1x	Indicates whether 802.1X authentication is enabled or disabled for each MAC address.

Displaying denied MAC addresses

Enter the **show table-mac-vlan denied-mac** command to display information about denied (authentication failed) MAC addresses.

```
Brocade(config)# show table-mac-vlan denied-mac
------------------------------------------------------------------------------
MAC Address     Port       Vlan Authenticated Time    Age   dot1x
------------------------------------------------------------------------------
0000.0030.1002   1/1/1       4092 No     00d00h11m57s H40    Dis
```

Syntax: show table-mac-vlan denied-mac

The following table describes the information in this output.

TABLE 51 Output field description of **show table-mac-vlan denied-mac** command

Field	Description
MAC Address	The denied MAC address for which the information is displayed.
Port	The port where MAC-based VLAN is enabled.
Vlan	This field displays VLAN 4092 for blocked hosts, or the restricted VLAN ID if it is configured on the port.
Authenticated	No indicates that authentication has failed. Inp indicates that authentication is in progress.
Time	The time at which authenticated failed.
Age	The age of the MAC address entry in the authenticated MAC address list.
Dot1x	Indicates whether 802.1X authentication is disabled (Dis) or enabled (Ena) for this MAC address.

Displaying detailed MAC-VLAN data

Enter the **show table-mac-vlan detailed** command to display a detailed version of MAC-VLAN information.

```
Brocade# show table-mac-vlan detailed  ethernet 1/1/2
Port                              : 1/1/2
Dynamic-Vlan  Assignment          : Disabled
RADIUS failure action             : Block Traffic
   Failure restrict use dot1x     : No
Override-restrict-vlan            : Yes
Vlan                              : (MAC-PERMIT-VLAN )
Port Vlan State                   : DEFAULT
802.1X override Dynamic PVID      : NO
Original PVID                     : 1
DOS attack protection             : Disabled
Accepted Mac Addresses            : 32
Rejected Mac Addresses            : 0
Authentication in progress        : 0
Authentication attempts           : 54
RADIUS timeouts                   : 16817
Num of MAC entries in TCAM        : 32
Num of MAC entries in MAC         : 32
Aging of MAC-sessions             : Enabled
Port move-back vlan               : Port-configured-vlan
Max-Age of sw mac session         : 60 seconds
hw age for denied mac             : 30 seconds
MAC Filter  applied               : No
--------------------------------------------------------------------------------
MAC Address    RADIUS       Authenticated  Time Age   CAM   MAC  Dot1x Type Pri
                                                      Index Index
--------------------------------------------------------------------------------
0000.0020.0012 0.0.0.0      No     00d00h00m00s S12   N/A   N/A  Dis   Dyn  0
0000.0020.0017 0.0.0.0      No     00d00h00m00s S20   N/A   N/A  Dis   Dyn  0
0000.0020.0018 0.0.0.0      No     00d00h00m00s S20   N/A   N/A  Dis   Dyn  0
0000.0020.000a 10.44.3.111  Yes    00d19h38m30s Ena   000b  22d4 Dis   Dyn  5
0000.0020.0019 0.0.0.0      No     00d00h00m00s S20   N/A   N/A  Dis   Dyn  0
0000.0020.001a 0.0.0.0      No     00d00h00m00s S20   N/A   N/A  Dis   Dyn  0
0000.0020.001b 0.0.0.0      No     00d00h00m00s S20   N/A   N/A  Dis   Dyn  0
0000.0020.001c 0.0.0.0      No     00d00h00m00s S20   N/A   N/A  Dis   Dyn  0
0000.0020.001d 0.0.0.0      No     00d00h00m00s S20   N/A   N/A  Dis   Dyn  0
--------------------------------------------------------------------------------
MAC Address    RADIUS       Authenticated  Time Age   CAM   MAC  Dot1x Type Pri
                                                      Index Index
--------------------------------------------------------------------------------
0000.00ed.1111 0.0.0.0      No     07d17h00m43s S0    0000  4000 Dis   Sta  1
0000.00ed.1112 0.0.0.0      No     07d17h01m51s S0    0001  4000 Dis   Sta  2
0000.00ed.1113 0.0.0.0      No     07d17h03m00s S0    0002  4000 Dis   Sta  3
```

Displaying MAC-VLAN information for a specific interface

Enter the **show table-mac-vlan ethernet** command to display MAC-VLAN information for a specific interface.

```
Brocade# show table-mac-vlan ethernet 1/1/1
------------------------------------------------------------------------------
MAC Address    Port    Vlan Authenticated  Time Age   CAM   MAC  Dot1x Type Pri
                                                      Index Index
------------------------------------------------------------------------------
0000.0010.0001 1/1/1   1    Yes    00d19h38m29s Ena   0008  0970 Dis   Dyn  0
0000.0010.0002 1/1/1   1    Yes    00d19h38m29s Ena   0009  0a40 Dis   Dyn  1
0000.0010.0003 1/1/1   1    Yes    00d19h38m30s Ena   000a  2b44 Dis   Dyn  2
0000.0010.0004 1/1/1   1    Yes    00d19h38m49s S96   0013  4000 Dis   Dyn  3
0000.0010.0005 1/1/1   1    Yes    00d19h38m53s Ena   0014  2d24 Dis   Dyn  4
0000.0010.0006 1/1/1   1    Yes    00d19h38m53s Ena   0015  2e14 Dis   Dyn  5
0000.0010.0007 1/1/1   1    Yes    00d19h38m41s S80   000f  4000 Dis   Dyn  6
0000.0010.0008 1/1/1   1    Yes    00d19h39m07s Ena   001f  00e0 Dis   Dyn  7
0000.0010.000a 1/1/1   1    Yes    00d19h38m30s Ena   000b  22d4 Dis   Dyn  0
0000.0010.0009 1/1/1   1    Yes    00d19h38m19s Ena   0001  21e4 Dis   Dyn  0
0000.0010.000a 1/1/1   1    Yes    00d19h38m30s Ena   000b  22d4 Dis   Dyn  0
0000.0010.000b 1/1/1   1    Yes    00d19h38m19s Ena   0002  03d0 Dis   Dyn  0
0000.0010.000c 1/1/1   1    Yes    00d19h38m57s Ena   001a  24b4 Dis   Dyn  0
0000.0010.000d 1/1/1   1    Yes    00d19h38m19s Ena   0003  05b0 Dis   Dyn  0
0000.0010.000e 1/1/1   1    Yes    00d19h38m31s S120  000c  4000 Dis   Dyn  0
0000.0010.000f 1/1/1   1    Yes    00d19h38m20s Ena   0004  2784 Dis   Dyn  0
0000.0010.0010 1/1/1   1    Yes    00d19h39m04s S32   001d  4000 Dis   Dyn  0
0000.0010.0011 1/1/1   1    Yes    00d19h38m43s Ena   0010  3864 Dis   Dyn  0
0000.0010.0012 1/1/1   1    Yes    00d19h38m39s Ena   000d  3b54 Dis   Dyn  0
```

The following table describes the information in this output.

TABLE 52 Output field descripion of the **show table-mac-vlan ethernet** command

Field	Description
MAC Address	The MAC addresses related to the specified interface.
Port	The interface for which this information is displayed.
Vlan	The VLAN to which the interface has been assigned.
Authenticated	Yes indicates authentication is successful. No indicates authentication has failed. Inp indicates authentication in progress Rst indicates a restricted VLAN
Time	The time at which the MAC address was authenticated. If the clock is set on the Brocade device, then the actual date and time are displayed. If the clock has not been set, then the time is displayed relative to when the device was last restarted.
Age	The age of the MAC address entry in the authenticated MAC address list.
CAM Index	This field displays the index of the CAM entry. The index value will be between 0 and 31. A value of “ff” indicates that the index is not used.
MAC Index	The index of the entry in the hardware MAC table.
Dot1x	Indicates whether 802.1X authentication is enabled or disabled for this MAC address.

TABLE 52 Output field descripion of the **show table-mac-vlan ethernet** command (Continued)

Field	Description
Type	Dyn Indicates a dynamic host. Sta indicates a static host.
Pri	This field indicates the value set for Foundry-MAC-based VLAN-QoS attribute in the RADIUS configuration for dynamic hosts, if configured. If the Foundry-MAC-based VLAN-QoS attribute is not configured, the value will be zero. For static hosts, the user-configured priority value for the MAC address is displayed.

Displaying MAC addresses in a MAC-based VLAN

Enter the **show mac-address** command to display a list of MAC addresses in a MAC-based VLAN.

```
Brocade# show mac-address
Total active entries from all ports = 1541
MAC-Address     Port     Type          Index    VLAN
0000.0020.0001  1/1/3 Dynamic(MBV)  1048       1
0000.0020.0002  1/1/3   Dynamic(MBV)  1832       1
0000.0020.0003  1/1/3   Dynamic(MBV)  9772       1
0000.0020.0004  1/1/3   Static(MBV)   328        1
0000.0020.0005  1/1/3   Dynamic(MBV)  8268       1
0000.0020.0006  1/1/3   Dynamic(MBV)  9084       1
0000.0020.0007  1/1/3   Dynamic(MBV)  632        1
0000.0020.0008  1/1/3   Dynamic(MBV)  3464       1
0000.0020.0009  1/1/3   Dynamic(MBV)  11404      1
0000.0020.000a  1/1/3   Dynamic(MBV)  12220      1
0000.0020.000b  1/1/3   Dynamic(MBV)  3768       1
```

NOTE
In this output, (MBV) indicates MAC-based VLAN is enabled.

The following table describes the output from this command.

TABLE 53 Output field description of the **show mac-address** command

Field	Description
Total active entries	The total number of active entries for all ports.
MAC Address	The MAC addresses assigned to this VLAN.
Port	The interface for which this information is displayed.
Type	Dynamic (MBV) Indicates a dynamic host. Static (MBV) indicates a static host.
Index	The index of the entry in the hardware MAC table.
VLAN	The VLAN to which these addresses are assigned.

Displaying MAC-based VLAN logging

Enter the **show logging** command to display MAC-based VLAN logging activity.

```
Brocade# show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 15 overruns)
    Buffer logging: level ACDMEINW, 50 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
Static Log Buffer
0d00h00m12s:A:System:  Power supply 1  is up
Dynamic Log Buffer (50 lines):
0d18h46m28s:I:running-config was changed from console
0d02h12m25s:A:MAC Based Vlan Mapping failed for [0000.110011.0108 ] on port 1/2/1
(Invalid User)
0d02h08m52s:A:MAC Based Vlan Mapping failed for [0000.0011.011b ] on port 1/2/1
(Invalid User)
0d02h05m01s:A:MAC Based Vlan Mapping failed for [0000.0011.00df ] on port 1/2/1
(Invalid User)
0d02h01m15s:A:MAC Based Vlan Mapping failed for [0000.0011.0108 ] on port 1/2/1
(Invalid User)
0d02h01m15s:A:MAC Based Vlan Mapping failed for [0000.0011.0107 ] on port 1/2/1
(Invalid User)
0d01h58m43s:N:MAC Based Vlan Enabled on port 1/2/1
0d01h58m32s:N:MAC Based Vlan Disabled on port 1/2/1
0d01h39m00s:I:running-config was changed from console
0d01h38m28s:I:System: Interface ethernet 1/1/7, state up
0d01h38m27s:I:System: Interface ethernet 1/1/6, state up
0d01h38m27s:I:System: Interface ethernet 1/1/4, state up
0d01h38m27s:I:System: Interface ethernet 1/1/5, state up
```

Clearing MAC-VLAN information

Enter the **clear table-mac-vlan** *interface* command to clear MAC-VLAN information. Add the interface id to clear information for a specific interface.

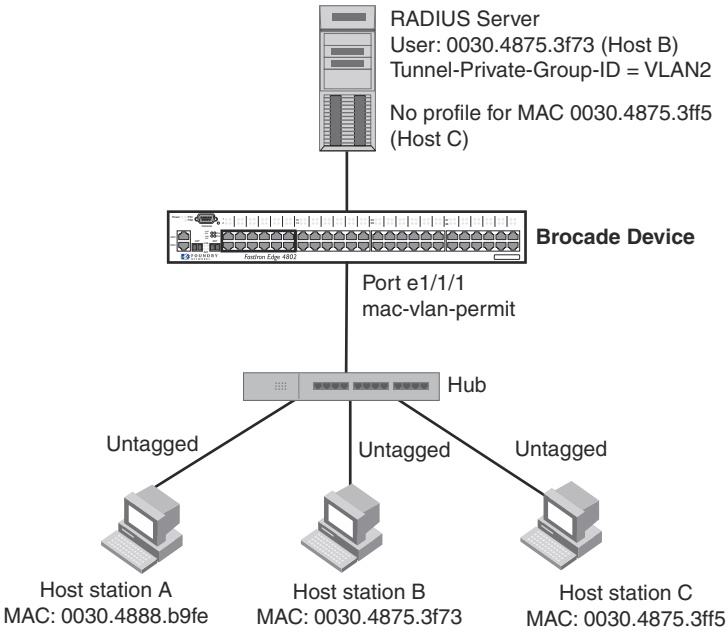
```
Brocade# clear table-mac-vlan <interface>
```

Sample MAC-based VLAN application

Figure 9 illustrates a sample configuration that uses MAC-based VLAN on port e 1/1/1 on the Brocade device. In this configuration, three host PCs are connected to port e 1/1/1 through a hub.

Host A MAC address is statically configured on port e 1/1/1. The profile for Host B MAC address on the RADIUS server specifies that the PC should be assigned to VLAN 2. Host C profile does not exist in the RADIUS server, and will be put into a restricted VLAN.

FIGURE 9 Sample MAC-based VLAN configuration



Host A MAC address is statically mapped to VLAN 1 with priority 1 and is not subjected to RADIUS authentication. When Host B MAC address is authenticated, the Access-Accept message from the RADIUS server specifies that Host B MAC address be placed into VLAN 2. Since Host C MAC address is not present in the RADIUS server, Host C will be rejected by the server and its MAC address will be placed into a restricted VLAN.

Below is the configuration for this example.

```
module 1 icx6650-64-56-port-management-module
module 2 icx6650-64-4-port-160g-module
module 3 icx6650-64-8-port-80g-module
vlan 1 by port
 untagged ethe 1/1/10
 mac-vlan-permit ethe 1/1/1 to 1/1/2
 no spanning-tree
vlan 2 by port
 untagged ethe 1/1/30
 mac-vlan-permit ethe 1/1/1 to 1/1/2
 no spanning-tree
vlan 666 name mac_restricted by port
 untagged ethe 1/1/20
 mac-vlan-permit ethe 1/1/1 to 1/1/2
 no spanning-tree
vlan 4000 name DEFAULT-VLAN by port
 no spanning-tree
vlan 4004 by port
 mac-vlan-permit ethe 1/1/1
default-vlan-id 4000
ip address 10.44.3.8 255.255.255.0
ip default-gateway 10.44.3.1
radius-server host 10.44.3.111
radius-server key 1 $-ndUno
mac-authentication enable
```

```
mac-authentication max-age 60
mac-authentication hw-deny-age 30
mac-authentication auth-passwd-format xxxx.xxxx.xxxx
interface ethernet 1/1/1
 mac-authentication mac-vlan max-mac-entries 5
 mac-authentication mac-vlan 0000.0088.b9fe vlan 1 priority 1
 mac-authentication mac-vlan enable
!
interface ethernet 1/1/2
mac-authentication mac-vlan max-mac-entries 5
mac-authentication mac-vlan enable
!
!
end
```

The **show table-mac-vlan** command returns the following results for all ports in this configuration.

```
Brocade# show table-mac-vlan
-----------------------------------------------------------------
Port   Vlan  Accepted  Rejected  Attempted  Static  Static  Max
             Macs      Macs      Macs       Macs    Conf    Macs
-----------------------------------------------------------------
1/1/1   N/A  2         1         0          1       1       5
1/1/2   N/A  0         0         0          0       0       5
```

The **show table-mac-vlan ethernet 1/1/1** command returns the following results for port1/1/1 in this configuration.

```
Brocade# show table-mac-vlan  ethernet 1/1/1
-------------------------------------------------------------------------------
MAC Address    Port    Vlan Authenticated  Time Age   CAM   MAC  Dot1x Type Pri
                                                      Index Index
-------------------------------------------------------------------------------
0000.0075.3f73 1/1/1   2    Yes    00d00h00m46s S32   0001  3728 Dis   Dyn  4
0000.0088.b9fe 1/1/1   1    Yes    00d00h00m08s Dis   0000  0970 Dis   Sta  1
0000.0075.3ff5 1/1/1   666  Rst    01d18h47m58s S8    0002  1ee4 Dis   Dyn  0
```

Chapter

9

Multi-Device Port Authentication

Table 54 lists the multi-device port authentication features supported on Brocade ICX 6650. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 54 Supported Multi-device port authentication (MDPA) features

Feature	Brocade ICX 6650
Multi-Device Port Authentication	Yes
Support for Multi-Device Port Authentication together with:	
• Dynamic VLAN assignment	Yes
• Dynamic ACLs	Yes
• 802.1X	Yes
• Denial of Service (DoS) attack protection	Yes
• Source guard protection	Yes
• ACL-per-port-per-VLAN	Yes
Authenticating multiple MAC addresses on an interface	Yes
Specifying the format of the MAC addresses sent to the RADIUS server	Yes
Specifying the authentication-failure action	Yes
Password override	Yes
Specifying the RADIUS timeout action	Yes
SNMP Traps	Yes
MAC Address Filters	Yes
Aging time for blocked MAC Addresses	Yes

Multi-device port authentication is a way to configure a Brocade device to forward or block traffic from a MAC address based on information received from a RADIUS server.

How multi-device port authentication works

Multi-device port authentication is a way to configure a Brocade device to forward or block traffic from a MAC address based on information received from a RADIUS server.

The multi-device port authentication feature 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 a RADIUS server. The MAC address itself is used as the username and password for RADIUS authentication; the user does not need to provide a specific username and password to gain access to the network. If RADIUS authentication for the MAC address is successful, traffic from the MAC address is forwarded in hardware.

If the RADIUS server cannot validate the user's MAC address, then it is considered an authentication failure, and a specified authentication-failure action can be taken. The default authentication-failure action is to drop traffic from the non-authenticated MAC address in hardware. You can also configure the device to move the port on which the non-authenticated MAC address was learned into a restricted or “guest” VLAN, which may have limited access to the network.

RADIUS authentication

The multi-device port authentication feature communicates with the RADIUS server to authenticate a newly found MAC address. The Brocade 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, 3 seconds) the RADIUS session times out, and the device retries the request up to three times. If no response is received, the next RADIUS server is chosen, and the request is sent for authentication.

The RADIUS server is configured with the usernames and passwords of authenticated users. For multi-device port authentication, the username and password is the MAC address itself; that is, the device uses the MAC address for both the username and the password in the request sent to the RADIUS server. For example, given a MAC address of 0007e90feaa1, the users file on the RADIUS server would be configured with a username and password both set to 0007e90feaa1. When traffic from this MAC address is encountered on a MAC-authentication-enabled interface, the device sends the RADIUS server an Access-Request message with 0007e90feaa1 as both the username and password. The format of the MAC address sent to the RADIUS server is configurable through the CLI.

The request for authentication from the RADIUS server is successful only if the username and password provided in the request matches an entry in the users database on the RADIUS server. When this happens, the RADIUS server returns an Access-Accept message back to the Brocade device. When the RADIUS server returns an Access-Accept message for a MAC address, that MAC address is considered authenticated, and traffic from the MAC address is forwarded normally by the Brocade device.

Authentication-failure actions

If the MAC address does not match the username and password of an entry in the users database on the RADIUS server, then the RADIUS server returns an Access-Reject message. When this happens, it is considered an authentication failure for the MAC address. When an authentication failure occurs, the Brocade device can either drop traffic from the MAC address in hardware (the default), or move the port on which the traffic was received to a restricted VLAN.

Supported RADIUS attributes

Brocade devices support the following RADIUS attributes for multi-device port authentication:

- Username (1) – RFC 2865
- NAS-IP-Address (4) – RFC 2865
- NAS-Port (5) – RFC 2865
- Service-Type (6) – RFC 2865
- FilterId (11) – RFC 2865
- Framed-MTU (12) – RFC 2865
- State (24) – RFC 2865
- Vendor-Specific (26) – RFC 2865
- Session-Timeout (27) – RFC 2865
- Termination-Action (29) – RFC 2865
- Calling-Station-ID (31) – RFC 2865
- NAS-Port-Type (61) – RFC 2865
- Tunnel-Type (64) – RFC 2868
- Tunnel-Medium-Type (65) – RFC 2868
- EAP Message (79) – RFC 2579
- Message-Authenticator (80) RFC 3579
- Tunnel-Private-Group-Id (81) – RFC 2868
- NAS-Port-id (87) – RFC 2869

Support for dynamic VLAN assignment

The Brocade multi-device port authentication feature supports ***dynamic VLAN assignment***, where a port can be placed in one or more VLANs based on the MAC address learned on that interface. For details about this feature, refer to "Configuring the RADIUS server to support dynamic VLAN assignment" on page 241.

Support for dynamic ACLs

The multi-device port authentication feature supports the assignment of a MAC address to a specific ACL, based on the MAC address learned on the interface. For details about this feature, refer to "Dynamically applying IP ACLs to authenticated MAC addresses" on page 243.

Support for authenticating multiple MAC addresses on an interface

The multi-device port authentication feature allows multiple MAC addresses to be authenticated or denied authentication on each interface. The maximum number of MAC addresses that can be authenticated on each interface is limited only by the amount of system resources available on the Brocade device.

Support for dynamic ARP inspection with dynamic ACLs

Multi-device port authentication and Dynamic ARP Inspection (DAI) are supported in conjunction with dynamic ACLs. Support is available in the Layer 3 software images only.

DAI is supported together with multi-device port authentication as long as **ACL-per-port-per-vlan** is enabled. Otherwise, you do not need to perform any extra configuration steps to enable support with dynamic ACLs. When these features are enabled on the same port/VLAN, support is automatically enabled.

Support for DHCP snooping with dynamic ACLs

Multi-device port authentication and DHCP snooping are supported in conjunction with dynamic ACLs. Support is available in the Layer 3 software images only.

DHCP Snooping is supported together with multi-device port authentication as long as **ACL-per-port-per-vlan** is enabled. Otherwise, you do not need to perform any extra configuration steps to enable support with dynamic ACLs. When these features are enabled on the same port/VLAN, support is automatically enabled.

Support for source guard protection

The Brocade proprietary ***Source Guard Protection*** feature, a form of IP Source Guard, can be used in conjunction with multi-device port authentication. For details, refer to "Enabling source guard protection" on page 246.

Multi-device port authentication and 802.1X security on the same port

On Brocade ICX 6650, multi-device port authentication and 802.1X security can be configured on the same port, as long as the port is not a trunk port or an LACP port. When both of these features are enabled on the same port, multi-device port authentication is performed prior to 802.1X authentication. If multi-device port authentication is successful, 802.1X authentication may be performed, based on the configuration of a vendor-specific attribute (VSA) in the profile for the MAC address on the RADIUS server.

NOTE
When multi-device port authentication and 802.1X security are configured together on the same port, Brocade recommends that dynamic VLANs and dynamic ACLs are done at the multi-device port authentication level, and not at the 802.1X level.

When both features are configured on a port, a device connected to the port is authenticated as follows.

1. Multi-device port authentication is performed on the device to authenticate the device MAC address.
2. If multi-device port authentication is successful for the device, then the device checks whether the RADIUS server included the Foundry-802_1x-enable VSA (described in Table 55) in the Access-Accept message that authenticated the device.
3. If the Foundry-802_1x-enable VSA is not present in the Access-Accept message, or is present and set to 1, then 802.1X authentication is performed for the device.

4. If the Foundry-802_1x-enable VSA is present in the Access-Accept message, and is set to 0, then 802.1X authentication is skipped. The device is authenticated, and any dynamic VLANs specified in the Access-Accept message returned during multi-device port authentication are applied to the port.
5. If 802.1X authentication is performed on the device, and is successful, then dynamic VLANs or ACLs specified in the Access-Accept message returned during 802.1X authentication are applied to the port.

If multi-device port authentication fails for a device, then by default traffic from the device is either blocked in hardware, or the device is placed in a restricted VLAN. You can optionally configure the Brocade device to perform 802.1X authentication on a device when it fails multi-device port authentication. Refer to "Example 2 — Creating a profile on the RADIUS server for each MAC address" on page 265 for a sample configuration where this is used.

Configuring Brocade-specific attributes on the RADIUS server

If the RADIUS authentication process is successful, the RADIUS server sends an Access-Accept message to the Brocade device, authenticating the device. The Access-Accept message can include Vendor-Specific Attributes (VSAs) that specify additional information about the device. If you are configuring multi-device port authentication and 802.1X authentication on the same port, then you can configure the Brocade VSAs listed in Table 55 on the RADIUS server.

You add these Brocade vendor-specific attributes to your RADIUS server configuration, and configure the attributes in the individual or group profiles of the devices that will be authenticated. The Brocade Vendor-ID is 1991, with Vendor-Type 1.

TABLE 55 Brocade vendor-specific attributes for RADIUS

Attribute name	Attribute ID	Data type	Description
Foundry-802_1x-enable	6	integer	Specifies whether 802.1X authentication is performed when multi-device port authentication is successful for a device. This attribute can be set to one of the following: **0** - Do not perform 802.1X authentication on a device that passes multi-device port authentication. Set the attribute to zero for devices that do not support 802.1X authentication. **1** - Perform 802.1X authentication when a device passes multi-device port authentication. Set the attribute to one for devices that support 802.1X authentication.
Foundry-802_1x-valid	7	integer	Specifies whether the RADIUS record is valid only for multi-device port authentication, or for both multi-device port authentication and 802.1X authentication. This attribute can be set to one of the following: **0** - The RADIUS record is valid only for multi-device port authentication. Set this attribute to zero to prevent a user from using their MAC address as username and password for 802.1X authentication **1** - The RADIUS record is valid for both multi-device port authentication and 802.1X authentication.

If neither of these VSAs exist in a device profile on the RADIUS server, then by default the device is subject to multi-device port authentication (if configured), then 802.1X authentication (if configured). The RADIUS record can be used for both multi-device port authentication and 802.1X authentication.

Configuration examples are shown in "Examples of multi-device port authentication and 802.1X authentication configuration on the same port" on page 263.

Multi-device port authentication configuration

Configuring multi-device port authentication on the Brocade device consists of the following tasks:

- Enabling multi-device port authentication globally and on individual interfaces
- Specifying the format of the MAC addresses sent to the RADIUS server (optional)
- Specifying the authentication-failure action (optional)
- Enabling and disabling SNMP traps for multi-device port authentication
- Defining MAC address filters (optional)
- Configuring dynamic VLAN assignment (optional)
- Dynamically Applying IP ACLs to authenticated MAC addresses
- Enabling denial of service attack protection (optional)

- Clearing authenticated MAC addresses (optional)
- Disabling aging for authenticated MAC addresses (optional)
- Configuring the hardware aging period for blocked MAC addresses
- Specifying the aging time for blocked MAC addresses (optional)

Enabling multi-device port authentication

To enable multi-device port authentication, you first enable the feature globally on the device. On some Brocade devices, you can then enable the feature on individual interfaces.

Globally enabling multi-device port authentication

To globally enable multi-device port authentication on the device, enter the following command.

```
Brocade(config)# mac-authentication enable
```

Syntax: [no] **mac-authentication enable**

Enabling multi-device port authentication on an interface

To enable multi-device port authentication on an individual interface, enter a command such as the following.

```
Brocade(config)# mac-authentication enable ethernet 1/3/1
```

Syntax: [no] **mac-authentication enable** *port* | **all**

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

The **all** option enables the feature on all interfaces at once.

You can enable the feature on an interface at the interface CONFIG level.

Example of enabling multi-device port authentication on an interface

```
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# mac-authentication enable
```

Syntax: [no] **mac-authentication enable**

You can also configure multi-device port authentication commands on a range of interfaces.

Example of enabling multi-device port authentication on a range of interfaces

```
Brocade(config)# internet ethernet 1/3/1 to 1/3/5
Brocade(config-mif-1/3/1-1/3/5)# mac-authentication enable
```

Specifying the format of the MAC addresses sent to the RADIUS server

When multi-device port authentication is configured, the Brocade device authenticates MAC addresses by sending username and password information to a RADIUS server. The username and password is the MAC address itself; that is, the device uses the MAC address for both the username and the password in the request sent to the RADIUS server.

By default, the MAC address is sent to the RADIUS server in the format *xxxxxxxxxxxx*. You can optionally configure the device to send the MAC address to the RADIUS server in the format *xx-xx-xx-xx-xx-xx*, or the format *xxxx.xxxx.xxxx*. To do this, enter a command such as the following.

```
Brocade(config)# mac-authentication auth-passwd-format xxxx.xxxx.xxxx
```

Syntax: [no] **mac-authentication auth-passwd-format xxxx.xxxx.xxxx** | **xx-xx-xx-xx-xx-xx** | **xxxxxxxxxxxx**

Specifying the authentication-failure action

When RADIUS authentication for a MAC address fails, you can configure the device to perform one of two actions:

- Drop traffic from the MAC address in hardware (the default)
- Move the port on which the traffic was received to a restricted VLAN

To configure the device to move the port to a restricted VLAN when multi-device port authentication fails, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# mac-authentication auth-fail-action
restrict-vlan 100
```

Syntax: [no] **mac-authentication auth-fail-action restrict-vlan** [*vlan-id*]

If the ID for the restricted VLAN is not specified at the interface level, the global restricted VLAN ID applies for the interface.

To specify the VLAN ID of the restricted VLAN globally, enter the following command.

```
Brocade(config)# mac-authentication auth-fail-vlan-id 200
```

Syntax: [no] **mac-authentication auth-fail-vlan-id** *vlan-id*

The command above applies globally to all MAC-authentication-enabled interfaces.

Note that the restricted VLAN must already exist on the device. You cannot configure the restricted VLAN to be a non-existent VLAN. If the port is a tagged or dual-mode port, you cannot use a restricted VLAN as the authentication-failure action.

To configure the device to drop traffic from non-authenticated MAC addresses in hardware, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# mac-authentication auth-fail-action
block-traffic
```

Syntax: [no] **mac-authentication auth-fail-action block-traffic**

Dropping traffic from non-authenticated MAC addresses is the default behavior when multi-device port authentication is enabled.

Generating traps for multi-device port authentication

You can enable and disable SNMP traps for multi-device port authentication. SNMP traps are enabled by default.

To enable SNMP traps for multi-device port authentication after they have been disabled, enter the following command.

```
Brocade(config)# snmp-server enable traps mac-authentication
```

Syntax: [no] **snmp-server enable traps mac-authentication**

Use the **no** form of the command to disable SNMP traps for multi-device port authentication.

Defining MAC address filters

You can specify MAC addresses that do not have to go through multi-device port authentication. These MAC addresses are considered pre-authenticated, and are not subject to RADIUS authentication. To do this, you can define MAC address filters that specify the MAC addresses to exclude from multi-device port authentication.

You should use a MAC address filter when the RADIUS server itself is connected to an interface where multi-device port authentication is enabled. If a MAC address filter is not defined for the MAC address of the RADIUS server and applied on the interface, the RADIUS authentication process would fail since the device would drop all packets from the RADIUS server itself.

For example, the following command defines a MAC address filter for address 0000.0058.aca4.

```
Brocade(config)# mac-authentication mac-filter 1 0000.0058.aca4
```

Syntax: [no] **mac-authentication mac-filter** *filter*

The following commands apply the MAC address filter on an interface so that address 0000.0058.aca4 is excluded from multi-device port authentication.

```
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# mac-authentication apply-mac-auth-filter 1
```

Syntax: [no] **mac-authentication apply-mac-auth-filter** *filter-id*

Configuring dynamic VLAN assignment

An interface can be dynamically assigned to one or more VLANs based on the MAC address learned on that interface. When a MAC address is successfully authenticated, the RADIUS server sends the Brocade device a RADIUS Access-Accept message that allows the Brocade device to forward traffic from that MAC address. The RADIUS Access-Accept message can also contain attributes set for the MAC address in its access profile on the RADIUS server.

If one of the attributes in the Access-Accept message specifies one or more VLAN identifiers, and the VLAN is available on the Brocade device, the port is moved from its default VLAN to the specified VLAN.

To enable dynamic VLAN assignment for authenticated MAC addresses, you must add attributes to the profile for the MAC address on the RADIUS server, then enable dynamic VLAN assignment on multi-device port authentication-enabled interfaces. Refer to "Configuring the RADIUS server to support dynamic VLAN assignment" on page 241 for a list of the attributes that must be set on the RADIUS server.

To enable dynamic VLAN assignment on a multi-device port authentication-enabled interface, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# mac-authentication enable-dynamic-vlan
```

Syntax: [no] mac-authentication enable-dynamic-vlan

Configuring a port to remain in the restricted VLAN after a successful authentication attempt

If a previous authentication attempt for a MAC address failed, and as a result the port was placed in the restricted VLAN, but a subsequent authentication attempt was successful, the RADIUS Access-Accept message may specify a VLAN for the port. By default, the Brocade device moves the port out of the restricted VLAN and into the RADIUS-specified VLAN. You can optionally configure the device to leave the port in the restricted VLAN. To do this, enter the following command.

```
Brocade(config-if-e10000-1/3/1)# mac-authentication no-override-restrict-vlan
```

When the above command is applied, if the RADIUS-specified VLAN configuration is tagged (e.g., T:1024) and the VLAN is valid, then the port is placed in the RADIUS-specified VLAN as a tagged port and left in the restricted VLAN. If the RADIUS-specified VLAN configuration is untagged (e.g., U:1024), the configuration from the RADIUS server is ignored, and the port is left in the restricted VLAN.

Syntax: [no] mac-authentication no-override-restrict-vlan

Configuration notes for configuring a port to remain in the restricted VLAN

- If you configure dynamic VLAN assignment on a multi-device port authentication enabled interface, and the Access-Accept message returned by the RADIUS server contains a Tunnel-Type and Tunnel-Medium-Type, but does not contain a Tunnel-Private-Group-ID attribute, then it is considered an authentication failure, and the configured authentication failure action is performed for the MAC address.
- If the *vlan-name* string does not match either the name or the ID of a VLAN configured on the device, then it is considered an authentication failure, and the configured authentication failure action is performed for the MAC address.
- For tagged or dual-mode ports, if the VLAN ID provided by the RADIUS server does not match the VLAN ID in the tagged packet that contains the authenticated MAC address as its source address, then it is considered an authentication failure, and the configured authentication failure action is performed for the MAC address.

- If an untagged port had previously been assigned to a VLAN through dynamic VLAN assignment, and then another MAC address is authenticated on the same port, but the RADIUS Access-Accept message for the second MAC address specifies a different VLAN, then it is considered an authentication failure for the second MAC address, and the configured authentication failure action is performed. Note that this applies only if the first MAC address has not yet aged out. If the first MAC address has aged out, then dynamic VLAN assignment would work as expected for the second MAC address.
- For dual mode ports, if the RADIUS server returns T:*vlan-name*, the traffic will still be forwarded in the statically assigned PVID. If the RADIUS server returns U:*vlan-name*, the traffic will not be forwarded in the statically assigned PVID.

Configuring the RADIUS server to support dynamic VLAN assignment

To specify VLAN identifiers on the RADIUS server, add the following attributes to the profile for the MAC address on the RADIUS server, then enable dynamic VLAN assignment on multi-device port authentication-enabled interfaces.

TABLE 56 Attributes for MAC address on RADIUS server

Attribute name	Type	Value
Tunnel-Type	064	13 (decimal) – VLAN
Tunnel-Medium-Type	065	6 (decimal) – 802
Tunnel-Private-Group-ID	081	*vlan-name*(string) The *vlan-name* value can specify either the name or the number of one or more VLANs configured on the Brocade device.

For information about the attributes, refer to "Dynamic VLAN assignment for 802.1X port configuration" on page 166.

Also, refer to the example configuration of "Multi-device port authentication with dynamic VLAN assignment" on page 260.

Enabling dynamic VLAN support for tagged packets on non-member VLAN ports

By default, the Brocade device drops tagged packets that are received on non-member VLAN ports. This process is called ***ingress filtering***. Since the MAC address of the packets are not learned, authentication does not take place.

The Brocade device can authenticate clients that send tagged packets on non-member VLAN ports. This enables the Brocade device to add the VLAN dynamically. To enable support, enter the following command at the Interface level of the CLI.

```
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# mac-authentication disable-ingress-filtering
```

If the client MAC address is successfully authenticated and the correct VLAN attribute is sent by the RADIUS server, the MAC address will be successfully authenticated on the VLAN.

Syntax: mac-authentication disable-ingress-filtering

Configuration notes and limitations

- This feature works in conjunction with multi-device port authentication with dynamic VLAN assignment only. If this feature is not enabled, authentication works as in "Example 1—Multi-device port authentication with dynamic VLAN assignment" on page 262.
- The port on which ingress filtering is disabled must be tagged to a VLAN.
- If a host sends both tagged and untagged traffic, and ingress filtering is disabled on the port, the port must be configured as a dual-mode port.

Specifying to which VLAN a port is moved after its RADIUS-specified VLAN assignment expires

When a port is dynamically assigned to a VLAN through the authentication of a MAC address, and the MAC session for that address is deleted on the Brocade device, then by default the port is removed from its RADIUS-assigned VLAN and placed back in the VLAN where it was originally assigned.

A port can be removed from its RADIUS-assigned VLAN when any of the following occur:

- The link goes down for the port
- The MAC session is manually deleted with the **mac-authentication clear-mac-session** command
- The MAC address that caused the port to be dynamically assigned to a VLAN ages out

For example, say port 1/1/1 is currently in VLAN 100, to which it was assigned when MAC address 0007.eaa1.e90f was authenticated by a RADIUS server. The port was originally configured to be in VLAN 111. If the MAC session for address 0007.eaa1.e90f is deleted, then port 1/1/1 is moved from VLAN 100 back into VLAN 111.

You can optionally specify an alternate VLAN to which to move the port when the MAC session for the address is deleted. For example, to place the port in the restricted VLAN, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# mac-auth move-back-to-old-vlan
port-restrict-vlan
```

Syntax: [no] **mac-authentication move-back-to-old-vlan port-restrict-vlan** | **port-configured-vlan** | **system-default-vlan**

The **port-configured-vlan** keyword removes the port from its RADIUS-assigned VLAN and places it back in the VLAN where it was originally assigned. This is the default.

The **port-restrict-vlan** keyword removes the port from its RADIUS-assigned VLAN and places it in the restricted VLAN.

The **system-default-vlan** keyword removes the port from its RADIUS-assigned VLAN and places it in the DEFAULT-VLAN.

NOTE

When a MAC session is deleted, if the port is moved back to a VLAN that is different than the running-config file, the system will update the running-config file to reflect the changes. This will occur even if **mac-authentication save-dynamicvlan-to-config"** is not configured.

Automatic removal of dynamic VLAN assignments for MAC authenticated ports

By default, the Brocade device removes any association between a port and a dynamically-assigned VLAN when all authenticated MAC sessions for that tagged or untagged VLAN have expired on the port. Thus, RADIUS-specified VLAN assignments are not saved to the device running-config file. When the **show run** command is issued during a session, dynamically-assigned VLANs are not displayed, although they can be displayed with the **show vlan**, **show auth-mac-addresses detail**, and **show auth-mac-addresses authorized-mac** commands.

You can optionally configure the Brocade device to save the RADIUS-specified VLAN assignments to the device's running-config file. Refer to "Saving dynamic VLAN assignments to the running-config file", next.

Saving dynamic VLAN assignments to the running-config file

By default, dynamic VLAN assignments are not saved to the running-config file of the Brocade device. However, you can configure the device to do so by entering the following command.

```
Brocade(config)# mac-authentication save-dynamicvlan-to-config
```

When the above command is applied, dynamic VLAN assignments are saved to the running-config file and are displayed when the **show run** command is issued. Dynamic VLAN assignments can also be displayed with the **show vlan**, **show auth-mac-addresses detail**, and **show auth-mac-addresses authorized-mac** commands.

Syntax: [no] **mac-authentication save-dynamicvlan-to-config**

Dynamically applying IP ACLs to authenticated MAC addresses

The Brocade multi-device port authentication implementation supports the assignment of a MAC address to a specific ACL, based on the MAC address learned on the interface.

When a MAC address is successfully authenticated, the RADIUS server sends the Brocade device a RADIUS Access-Accept message that allows the Brocade device to forward traffic from that MAC address. The RADIUS Access-Accept message can also contain, among other attributes, the Filter-ID (type 11) attribute for the MAC address. When the Access-Accept message containing the Filter-ID (type 11) attribute is received by the Brocade device, it will use the information in these attributes to apply an IP ACL on a per-MAC (per user) basis.

The dynamic IP ACL is active as long as the client is connected to the network. When the client disconnects from the network, the IP ACL is no longer applied to the port. If an IP ACL had been applied to the port prior to multi-device port authentication; it will be re-applied to the port.

NOTE
A dynamic IP ACL will take precedence over an IP ACL that is bound to a port (port ACL). When a client authenticates with a dynamic IP ACL, the port ACL will not be applied. Also, future clients on the same port will authenticate with a dynamic IP ACL or no IP ACL. If no clients on the port use dynamic ACL, then the port ACL will be applied to all traffic.

The Brocade device uses information in the Filter ID to apply an IP ACL on a per-user basis. The Filter-ID attribute can specify the number of an existing IP ACL configured on the Brocade device. If the Filter-ID is an ACL number, the specified IP ACL is applied on a per-user basis.

Multi-device port authentication with dynamic IP ACLs and ACL-per-port-per-VLAN

Multi-device port authentication and dynamic ACLs are supported on tagged, dual-mode, and untagged ports, with or without virtual interfaces.

Support is automatically enabled when all of the required conditions are met.

The following describes the conditions and feature limitations:

- On Layer 3 router code, dynamic IP ACLs are allowed on physical ports when **ACL-per-port-per-vlan** is enabled.
- On Layer 3 router code, dynamic IP ACLs are allowed on tagged and dual-mode ports when **ACL-per-port-per-vlan** is enabled. If **ACL-per-port-per-vlan** is not enabled, dynamic IP ACLs are not allowed on tagged or dual-mode ports.
- Dynamic IP ACLs can be added to tagged/untagged ports in a VLAN with or without a VE, as long as the tagged/untagged ports do not have configured ACLs assigned to them. The following shows some example scenarios where dynamic IP ACLs would not apply:
  - A port is a tagged/untagged member of VLAN 20, VLAN 20 includes VE 20, and an ACL is bound to VE 20.
  - A port is a tagged/untagged member of VLAN 20, VLAN 20 includes VE 20, and a per-port-per-vlan ACL is bound to VE 20 and to a subset of ports in VE 20

  In the above scenarios, dynamic IP ACL assignment would not apply in either instance, because a configured ACL is bound to VE 20 on the port. Consequently, the MAC session would fail.

Configuration considerations and guidelines for multi-device port authentication

- Dynamic IP ACLs with multi-device port authentication are supported. Dynamic MAC address filters with multi-device port authentication are not supported.
- In the Layer 2 switch code, dynamic IP ACLs are not supported when **ACL-per-port-per-vlan** is enabled on a global-basis.
- The RADIUS Filter ID (type 11) attribute is supported. The Vendor-Specific (type 26) attribute is not supported.
- The dynamic ACL must be an extended ACL. Standard ACLs are not supported.
- Multi-device port authentication and 802.1x can be used together on the same port. However, Brocade does not recommend the use of multi-device port authentication and 802.1X with dynamic ACLs together on the same port. If a single supplicant requires both 802.1x and multi-device port authentication, and if both 802.1x and multi-device port authentication try to install different dynamic ACLs for the same supplicant, the supplicant will fail authentication.
- Dynamically assigned IP ACLs are subject to the same configuration restrictions as non-dynamically assigned IP ACLs. One caveat is that ports with VE interfaces cannot have assigned user-defined ACLs. For example, a user-defined ACL bound to a VE or a port on a VE is not allowed. There are no restrictions on ports that do not have VE interfaces.

- Dynamic ACL filters are supported only for the inbound direction. Dynamic outbound ACL filters are not supported.
- Dynamic ACL assignment with multi-device port authentication is not supported in conjunction with any of the following features:
  - IP source guard
  - Rate limiting
  - Protection against ICMP or TCP Denial-of-Service (DoS) attacks
  - Policy-based routing
  - 802.1X dynamic filter

Configuring the RADIUS server to support dynamic IP ACLs

When a port is authenticated using multi-device port authentication, an IP ACL filter that exists in the running-config file on the Brocade device can be dynamically applied to the port. To do this, you configure the Filter-ID (type 11) attribute on the RADIUS server. The Filter-ID attribute specifies the name or number of the Brocade IP ACL.

The following is the syntax for configuring the Filter-ID attribute on the RADIUS server to refer to a Brocade IP ACL.

TABLE 57 Syntax for configuring the Filter-ID attribute

Value	Description
ip.*number*.in[1]	Applies the specified numbered ACL to the authenticated port in the inbound direction.
ip.*name*.in[1,2]	Applies the specified named ACL to the authenticated port in the inbound direction.

1. The ACL must be an extended ACL. Standard ACLs are not supported.
2. The *name* in the Filter ID attribute is case-sensitive

The following table lists examples of values you can assign to the Filter-ID attribute on the RADIUS server to refer to IP ACLs configured on a Brocade device.

TABLE 58 Filter-ID values

Possible values for the filter ID attribute on the RADIUS server	ACLs configured on the Brocade device
ip.102.in	access-list 102 permit ip 36.0.0.0 0.255.255.255 any
ip.fdry_filter.in	ip access-list standard fdry_filter permit host 36.48.0.3

Enabling denial of service attack protection

The Brocade device does not start forwarding traffic from an authenticated MAC address in hardware until the RADIUS server authenticates the MAC address; traffic from the non-authenticated MAC addresses is sent to the CPU. A denial of service (DoS) attack could be launched against the device where a high volume of new source MAC addresses is sent to the device, causing the CPU to be overwhelmed with performing RADIUS authentication for these MAC addresses. In addition, the high CPU usage in such an attack could prevent the RADIUS response from reaching the CPU in time, causing the device to make additional authentication attempts.

To limit the susceptibility of the Brocade device to such attacks, you can configure the device to use multiple RADIUS servers, which can share the load when there are a large number of MAC addresses that need to be authenticated. The Brocade device can run a maximum of 10 RADIUS clients per server and will attempt to authenticate with a new RADIUS server if current one times out.

In addition, you can configure the Brocade device to limit the rate of authentication attempts sent to the RADIUS server. When the multi-device port authentication feature is enabled, it keeps track of the number of RADIUS authentication attempts made per second. When you also enable the DoS protection feature, if the number of RADIUS authentication attempts for MAC addresses learned on an interface per second exceeds a configurable rate (by default 512 authentication attempts per second), the device considers this a possible DoS attack and disables the port. You must then manually re-enable the port.

The DoS protection feature is disabled by default. To enable it on an interface, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# mac-authentication dos-protection enable
```

ITo specify a maximum rate for RADIUS authentication attempts, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# mac-authentication dos-protection mac-limit 256
```

Syntax: [**no**] **mac-authentication dos-protection mac-limit** *number*

You can specify a rate from 1–65535 authentication attempts per second. The default is a rate of 512 authentication attempts per second.

Enabling source guard protection

Source Guard Protection is a form of IP Source Guard used in conjunction with multi-device port authentication. When Source Guard Protection is enabled, IP traffic is blocked until the system learns the IP address. Once the IP address is validated, traffic with that source address is permitted.

NOTE
Source Guard Protection is supported together with multi-device port authentication as long as **ACL-per-port-per-vlan** is enabled.

When a new MAC session begins on a port that has Source Guard Protection enabled, the session will either apply a dynamically created Source Guard ACL entry, or it will use the dynamic IP ACL assigned by the RADIUS server. If a dynamic IP ACL is not assigned, the session will use the Source Guard ACL entry. The Source Guard ACL entry is **permit ip** *secure-ip* **any**, where *secure-ip* is obtained from the ARP Inspection table or from the DHCP Secure table. The DHCP Secure table is comprised of DHCP Snooping and Static ARP Inspection entries.

The Source Guard ACL permit entry is added to the hardware table after all of the following events occur:

- The MAC address is authenticated
- The IP address is learned

- The MAC-to-IP mapping is checked against the Static ARP Inspection table or the DHCP Secure table.

The Source Guard ACL entry is not written to the running configuration file. However, you can view the configuration using the **show auth-mac-addresses authorized-mac ip-addr**. Refer to "Viewing the assigned ACL for ports on which source guard protection is enabled" in the following section.

NOTE
The secure MAC-to-IP mapping is assigned at the time of authentication and remains in effect as long as the MAC session is active. If the DHCP Secure table is updated after the session is authenticated and while the session is still active, it does not affect the existing MAC session.

The Source Guard ACL permit entry is removed when the MAC session expires or is cleared.

To enable Source Guard Protection on a port on which multi-device port authentication is enabled, enter the following command at the Interface level of the CLI.

```
Brocade(config)# interface ethernet 1/1/4
Brocade(config-if-e10000-1/1/4)# mac-authentication source-guard-protection
enable
```

Syntax: [no] **mac-authentication source-guard-protection enable**

Enter the **no** form of the command to disable SG protection.

Viewing the assigned ACL for ports on which source guard protection is enabled

Use the following command to view whether a Source Guard ACL or dynamic ACL is applied to ports on which Source Guard Protection is enabled.

```
Brocade(config)# show auth-mac-addresses authorized-mac ip-addr
-------------------------------------------------------------------------------
MAC Address    SourceIp        Port   Vlan  Auth Age  ACL  dot1x
-------------------------------------------------------------------------------
0000.0010.2000 200.1.17.5      1/1/2    171   Yes Dis   SG  Ena
0000.0010.2001 200.1.17.6      1/1/3    171   Yes Dis  103  Ena
```

In the above output, for port 1/1/2, Source Guard Protection is enabled and the Source Guard ACL is applied to the MAC session, as indicated by **SG** in the **ACL** column. For port 1/1/3, Source Guard Protection is also enabled, but in this instance, a dynamic ACL (103) is applied to the MAC session.

Clearing authenticated MAC addresses

The Brocade device maintains an internal table of the authenticated MAC addresses (viewable with the **show authenticated-mac-address** command). You can clear the contents of the authenticated MAC address table either entirely, or just for the entries learned on a specified interface. In addition, you can clear the MAC session for an address learned on a specific interface.

To clear the entire contents of the authenticated MAC address table, enter the **clear auth-mac-table** command.

```
Brocade# clear auth-mac-table
```

Syntax: clear auth-mac-table

To clear the authenticated MAC address table of entries learned on a specified interface, enter a command such as the following.

```
Brocade# clear auth-mac-table ethernet 1/3/1
```

Syntax: clear auth-mac-table ethernet *port*

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

To clear the MAC session for an address learned on a specific interface, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# mac-authentication clear-mac-session
00e0.1234.abd4
```

Syntax: mac-authentication clear-mac-session *mac-address*

This command removes the Layer 2 CAM entry created for the specified MAC address. If the Brocade device receives traffic from the MAC address again, the MAC address is authenticated again.

NOTE
In a configuration with multi-device port authentication and 802.1X authentication on the same port, the **mac-authentication clear-mac-session** command will clear the MAC session, as well as its respective 802.1X session, if it exists.

Disabling aging for authenticated MAC addresses

MAC addresses that have been authenticated or denied by a RADIUS server are aged out if no traffic is received from the MAC address for a certain period of time:

- Authenticated MAC addresses or non-authenticated MAC addresses that have been placed in the restricted VLAN are aged out if no traffic is received from the MAC address over the device normal MAC aging interval.
- Non-authenticated MAC addresses that are blocked by the device are aged out if no traffic is received from the address over a fixed hardware aging period (70 seconds), plus a configurable software aging period. (Refer to the next section for more information on configuring the software aging period).

You can optionally disable aging for MAC addresses subject to authentication, either for all MAC addresses or for those learned on a specified interface.

Globally disabling aging of MAC addresses

On most devices, you can disable aging for all MAC addresses on all interfaces where multi-device port authentication has been enabled by entering the **mac-authentication disable-aging** command.

```
Brocade(config)# mac-authentication disable-aging
```

Syntax: mac-authentication disable-aging

Enter the command at the global or interface configuration level.

The **denied-only** parameter prevents denied sessions from being aged out, but ages out permitted sessions.

The **permitted-only** parameter prevents permitted (authenticated and restricted) sessions from being aged out and ages denied sessions.

Disabling the aging of MAC addresses on interfaces

To disable aging for all MAC addresses subject to authentication on a specific interface where multi-device port authentication has been enabled, enter the command at the interface level.

Example

```
Brocade(config)# interface ethernet 1/3/1
Brocade(config-if-e10000-1/3/1)# mac-authentication disable-aging
```

Syntax: [**no**] **mac-authentication disable-aging**

Changing the hardware aging period for blocked MAC addresses

When the Brocade device is configured to drop traffic from non-authenticated MAC addresses, traffic from the blocked MAC addresses is dropped in hardware, without being sent to the CPU. A Layer 2 hardware entry is created that drops traffic from the MAC address in hardware. If no traffic is received from the MAC address for a certain amount of time, this Layer 2 hardware entry is aged out. If traffic is subsequently received from the MAC address, then an attempt can be made to authenticate the MAC address again.

Aging of the Layer 2 hardware entry for a blocked MAC address occurs in two phases, known as hardware aging and software aging.

On Brocade ICX 6650 devices, the hardware aging period for blocked MAC addresses is fixed at 70 seconds and is non-configurable. (The hardware aging time for non-blocked MAC addresses is the length of time specified with the **mac-age** command.) The software aging period for blocked MAC addresses is configurable through the CLI, with the **mac-authentication max-age** command. After the hardware aging period ends, the software aging period begins. When the software aging period ends, the blocked MAC address ages out, and can be authenticated again if the Brocade device receives traffic from the MAC address.

To change the hardware aging period for blocked MAC addresses, enter a command such as the following.

```
Brocade(config)# mac-authentication hw-deny-age 10
```

Syntax: [**no**] **mac-authentication hw-deny-age** *num*

The *num* parameter is a value from 1 to 65535 seconds. The default is 70 seconds.

Specifying the aging time for blocked MAC addresses

When the Brocade device is configured to drop traffic from non-authenticated MAC addresses, traffic from the blocked MAC addresses is dropped in hardware, without being sent to the CPU. A Layer 2 CAM entry is created that drops traffic from the blocked MAC address in hardware. If no traffic is received from the blocked MAC address for a certain amount of time, this Layer 2 CAM entry is aged out. If traffic is subsequently received from the MAC address, then an attempt can be made to authenticate the MAC address again.

Aging of the Layer 2 CAM entry for a blocked MAC address occurs in two phases, known as ***hardware aging*** and ***software aging***. The hardware aging period is fixed at 70 seconds and is non-configurable. The software aging time is configurable through the CLI.

Once the Brocade device stops receiving traffic from a blocked MAC address, the hardware aging begins and lasts for a fixed period of time. After the hardware aging period ends, the software aging period begins. The software aging period lasts for a configurable amount of time (by default 120 seconds). After the software aging period ends, the blocked MAC address ages out, and can be authenticated again if the Brocade device receives traffic from the MAC address.

To change the length of the software aging period for blocked MAC addresses, enter a command such as the following.

```
Brocade(config)# mac-authentication max-age 180
```

Syntax: [no] **mac-authentication max-age** *seconds*

You can specify from 1–65535 seconds. The default is 120 seconds.

Specifying the RADIUS timeout action

A RADIUS timeout occurs when the Brocade device does not receive a response from a RADIUS server within a specified time limit and after a certain number of retries. The time limit and number of retries can be manually configured using the CLI commands **radius-server timeout** and **radius-server retransmit**, respectively. If the parameters are not manually configured, the Brocade device applies the default value of three seconds with a maximum of three retries.

You can better control port behavior when a RADIUS timeout occurs by configuring a port on the Brocade device to automatically pass or fail user authentication. A ***pass*** essentially bypasses the authentication process and permits user access to the network. A ***fail*** bypasses the authentication process and blocks user access to the network, unless restrict-vlan is configured, in which case, the user is placed into a VLAN with restricted or limited access. By default, the Brocade device will reset the authentication process and retry to authenticate the user.

Specify the RADIUS timeout action at the Interface level of the CLI.

Permit user access to the network after a RADIUS timeout

To set the RADIUS timeout behavior to bypass multi-device port authentication and *permit* user access to the network, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/1/3
Brocade(config-if-e10000-1/1/3)# mac-authentication auth-timeout-action success
```

Syntax: [no] **mac-authentication auth-timeout-action success**

Once the *success* timeout action is enabled, use the no form of the command to reset the RADIUS timeout behavior to *retry*.

Deny user access to the network after a RADIUS timeout

To set the RADIUS timeout behavior to bypass multi-device port authentication and block user access to the network, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/1/3
Brocade(config-if-e10000-1/1/3)# mac-authentication auth-timeout-action failure
```

Syntax: [**no**] **mac-authentication auth-timeout-action failure**

After the failure timeout action is enabled, use the **no** form of the command to reset the RADIUS timeout behavior to *retry*.

NOTE
If **restrict-vlan** is configured along with ***auth-timeout-action failure***, the user will be placed into a VLAN with restricted or limited access. Refer to "Allow user access to a restricted VLAN after a RADIUS timeout" on page 251.

Allow user access to a restricted VLAN after a RADIUS timeout

To set the RADIUS timeout behavior to bypass multi-device port authentication and place the user in a VLAN with restricted or limited access, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/1/3
Brocade(config-if-e10000-1/1/3)# mac-authentication auth-fail-action
restrict-vlan 100
Brocade(config-if-e10000-1/1/3)# mac-authentication auth-timeout-action failure
```

Syntax: [**no**] **mac-authentication auth-fail-action restrict-vlan** [*vlan-id*]

Syntax: [**no**] **mac-authentication auth-timeout-action failure**

Multi-device port authentication password override

The multi-device port authentication feature communicates with the RADIUS server to authenticate a newly found MAC address. The RADIUS server is configured with the usernames and passwords of authenticated users. For multi-device port authentication, the username and password is the MAC address itself; that is, the device uses the MAC address for both the username and the password in the request sent to the RADIUS server. For example, given a MAC address of 0000000feaa1, the users file on the RADIUS server would be configured with a username and password both set to 0000000feaa1. When traffic from this MAC address is encountered on a MAC-authentication-enabled interface, the device sends the RADIUS server an Access-Request message with 0000000feaa1 as both the username and password.

The MAC address is the *default* password for multi-device port authentication, and you can optionally configure the device to use a different password. Note that the MAC address is still the username and cannot be changed.

To change the password for multi-device port authentication, enter a command such as the following at the GLOBAL Config Level of the CLI.

```
Brocade(config)# mac-authentication password-override
```

Syntax: [no] **mac-authentication password-override** *password*

where *password* can have up to 32 alphanumeric characters, but cannot include blank spaces.

Limiting the number of authenticated MAC addresses

You cannot enable MAC port security on the same port that has multi-device port authentication enabled. To simulate the function of MAC port security, you can enter a command such as the following.

```
Brocade(config-if-e10000-1/1/2)# mac-authentication max-accepted-session 5
```

Syntax: [no] **mac-authentication max-accepted-session** *session-number*

This command limits the number of successfully authenticated MAC addresses. Enter a value from 1 - 250 for session-number

Displaying multi-device port authentication information

You can display the following information about the multi-device port authentication configuration:

- Information about authenticated MAC addresses
- Information about the multi-device port authentication configuration
- Authentication Information for a specific MAC address or port
- Multi-device port authentication settings and authenticated MAC addresses for each port where the multi-device port authentication feature is enabled
- The MAC addresses that have been successfully authenticated
- The MAC addresses for which authentication was not successful

Displaying authenticated MAC address information

To display information about authenticated MAC addresses on the ports where the multi-device port authentication feature is enabled, enter the show **auth-mac address** command.

```
Brocade# show auth-mac-address
------------------------------------------------------------------------
Port           Vlan  Accepted MACs    Rejected MACs    Attempted-MACs
------------------------------------------------------------------------
1/1/8           100   1                  100               0
1/2/1           40    0                  0                 0
1/2/2           100   0                  0                 0
1/3/5           30    0                  0                 0
```

Syntax: show auth-mac-address

The following table describes the information displayed by the **show auth-mac-address** command.

TABLE 59 Output from the **show authenticated-mac-address command**

Field	Description
Port	The port number where the multi-device port authentication feature is enabled.
Vlan	The VLAN to which the port has been assigned.
Accepted MACs	The number of MAC addresses that have been successfully authenticated
Rejected MACs	The number of MAC addresses for which authentication has failed.
Attempted-MACs	The rate at which authentication attempts are made for MAC addresses.

Displaying multi-device port authentication configuration information

To display information about the multi-device port authentication configuration, enter the **show auth-mac-address configuration** command.

```
Brocade# show auth-mac-address configuration

Feature enabled                 : Yes
Number of Ports enabled         : 4
-------------------------------------------------------------------------
Port   Fail-Action     Fail-vlan   Dyn-vlan  MAC-filter
-------------------------------------------------------------------------
1/1/8  Block Traffic    1            No        No
1/2/1  Block Traffic    1            No        No
1/2/2  Block Traffic    1            No        Yes
1/2/5  Block Traffic    1            No        No
```

Syntax: show auth-mac-address configuration

The following table describes the output from the **show auth-mac-address configuration** command.

TABLE 60 Output from the **show authenticated-mac-addres**s configuration c**ommand**

Field	Description
Feature enabled	Whether multi-device port authentication is enabled on the Brocade device.
Number of Ports enabled	The number of ports on which the multi-device port authentication feature is enabled.
Port	Information for each multi-device port authentication-enabled port.
Fail-Action	What happens to traffic from a MAC address for which RADIUS authentication has failed either block the traffic or assign the MAC address to a restricted VLAN.
Fail-vlan	The restricted VLAN to which non-authenticated MAC addresses are assigned, if the Fail-Action is to assign the MAC address to a restricted VLAN.
Dyn-vlan	Whether RADIUS dynamic VLAN assignment is enabled for the port.
MAC-filter	Whether a MAC address filter has been applied to specify pre-authenticated MAC addresses.

Displaying multi-device port authentication information for a specific MAC address or port

To display authentication information for a specific MAC address or port, enter a command such as the following.

```
Brocade# show auth-mac-address 0000.000f.eaa1
--------------------------------------------------------------------------------
MAC/IP Address                    Port        Vlan Authenticated Time  Age CAM
                                                                          Index
--------------------------------------------------------------------------------
0000.000f.eaa1 : 25.25.25.25      1/1/8        100  Yes    00d01h10m06s 0   N/A
```

Syntax: show auth-mac-address *mac-address* | *ip-addr* | *port*

The *ip-addr* variable lists the MAC address associated with the specified IP address.

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

The following table describes the information displayed by the **show authenticated-mac-address** command for a specified MAC address or port.

TABLE 61 Output from the **show authenticated-mac-addres**s ***address*** command

Field	Description
MAC/IP Address	The MAC address for which information is displayed. If the packet for which multi-device port authentication was performed also contained an IP address, then the IP address is displayed as well.
Port	The port on which the MAC address was learned.
Vlan	The VLAN to which the MAC address was assigned.
Authenticated	Whether the MAC address was authenticated.

TABLE 61 Output from the **show authenticated-mac-address *address*** command (Continued)

Field	Description
Time	The time at which the MAC address was authenticated. If the clock is set on the Brocade device, then the actual date and time are displayed. If the clock has not been set, then the time is displayed relative to when the device was last restarted.
Age	The age of the MAC address entry in the authenticated MAC address list.
CAM Index	If the MAC address is blocked, this is the index entry for the Layer 2 CAM entry created for this MAC address. If the MAC address is not blocked, either through successful authentication or through being placed in the restricted VLAN, then "N/A" is displayed. If the hardware aging period has expired, then "ffff" is displayed for the MAC address during the software aging period.

Displaying the authenticated MAC addresses

To display the MAC addresses that have been successfully authenticated, enter the **show auth-mac-addresses authorized-mac** command.

The display output on your device may differ, depending on the software version running on the device.

```
Brocade# show auth-mac-addresses authorized-mac
--------------------------------------------------------------------------------
MAC Address     Port        Vlan Authenticated Time  Age  dot1x
--------------------------------------------------------------------------------
0000.0074.3181   1/2/3       101   Yes 00d01h03m17s   Ena  Ena
0000.0000.0001   1/1/1        87   Yes 00d01h03m17s   Ena  Ena
0000.0000.012d   1/1/1        87   Yes 00d01h03m17s   Ena  Ena
0000.0000.0065   1/1/1        87   Yes 00d01h03m17s   Ena  Ena
0000.0000.0191   1/1/1        87   Yes 00d01h03m17s   Ena  Ena
0000.0000.01f5   1/1/1        87   Yes 00d01h03m17s   Ena  Ena
```

Syntax: show auth-mac-addresses authorized-mac

Displaying the non-authenticated MAC addresses

To display the MAC addresses for which authentication was not successful, enter the **show auth-mac-addresses unauthorized-mac** command

```
Brocade# show auth-mac-addresses unauthorized-mac
------------------------------------------------------------------------------
MAC Address     Port        Vlan Authenticated Time  Age  dot1x
------------------------------------------------------------------------------
0000.0000.0321   1/8/1        87   No  00d01h03m17s   H44  Ena
0000.0000.0259   1/8/1        87   No  00d01h03m17s   H44  Ena
0000.0000.0385   1/8/1        87   No  00d01h03m17s   H44  Ena
0000.0000.02bd   1/8/1        87   No  00d01h03m17s   H44  Ena
0000.0000.00c9   1/8/1        87   No  00d01h03m17s   H44  Ena
```

Syntax: show auth-mac-addresses unauthorized-mac

Table 62 explains the information in the output.

Displaying multi-device port authentication information for a port

To display a summary of Multi-Device Port Authentication for ports on a device, enter the following command

```
Brocade# show auth-mac-addresses ethernet 1/8/1
------------------------------------------------------------------------------
MAC Address      Port   Vlan Authenticated Time  Age Dot1x
------------------------------------------------------------------------------
0000.0000.0001    1/8/1   87   Yes 00d01h03m17s   Ena  Ena
0000.0000.012d    1/8/1   87   Yes 00d01h03m17s   Ena  Ena
0000.0000.0321    1/8/1   87   No  00d01h03m17s   H52  Ena
0000.0000.0259    1/8/1   87   No  00d01h03m17s   H52  Ena
0000.0000.0065    1/8/1   87   Yes 00d01h03m17s   Ena  Ena
0000.0000.0385    1/8/1   87   No  00d01h03m17s   H52  Ena
0000.0000.0191    1/8/1   87   Yes 00d01h03m17s   Ena  Ena
0000.0000.02bd    1/8/1   87   No  00d01h03m17s   H52  Ena
0000.0000.00c9    1/8/1   87   No  00d01h03m17s   H52  Ena
0000.0000.01f5    1/8/1   87   Yes 00d01h03m17s   Ena  Ena
```

Syntax: show auth-mac-address ethernet *port*

Table 62 explains the information in the output.

TABLE 62 Output of show auth-mac-address

Field	Description
MAC Address	The MAC addresses learned on the port. If the packet for which multi-device port authentication was performed also contained an IP address, the IP address is also displayed.
Port	ID of the port on which the MAC address was learned.
VLAN	VLAN of which the port is a member.

TABLE 62 Output of show auth-mac-address (Continued)

Field	Description
Authenticated	Whether the MAC address has been authenticated by the RADIUS server.
Time	The time the MAC address was authenticated. If the clock is set on the Brocade device, then the actual date and time are displayed. If the clock has not been set, the time is displayed relative to when the device was last restarted.
Age	The age of the MAC address entry in the authenticated MAC address list.
Dot1x	Indicates if 802.1X authentication is enabled or disabled for the MAC address

Displaying multi-device port authentication settings and authenticated MAC addresses

To display the multi-device port authentication settings and authenticated MAC addresses for a port where the feature is enabled, enter the following command.

Syntax: show auth-mac-address [**detail**] [**ethernet** *port*]

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

Omitting the **ethernet** *port* parameter displays information for all interfaces where the multi-device port authentication feature is enabled.

```
Brocade# show auth-mac-addresses detailed ethernet 1/2/3
Port                              : 1/2/3
Dynamic-Vlan  Assignment          : Enabled
RADIUS failure action             : Block Traffic
   Failure restrict use dot1x     : No
Override-restrict-vlan            : Yes
Port Default VLAN                 : 101 ( RADIUS assigned: No) (101)
Port Vlan State                   : DEFAULT
802.1x override Dynamic PVID      : YES
     override return to PVID      : 101
Original PVID                     : 101
DOS attack protection             : Disabled
Accepted Mac Addresses            : 1
Rejected Mac Addresses            : 0
Authentication in progress        : 0
Authentication attempts           : 0
RADIUS timeouts                   : 0
RADIUS timeouts action            : Success
MAC Address on PVID               : 1
MAC Address authorized on PVID    : 1
Aging of MAC-sessions             : Enabled
Port move-back vlan               : Port-configured-vlan
Max-Age of sw mac session         : 120 seconds
hw age for denied mac             : 70 seconds
MAC Filter  applied               : No
Dynamic ACL applied               : No
num Dynamic Tagged Vlan           : 2
Dynamic Tagged Vlan list          : 1025 (1/1) 4060 (1/0)

--------------------------------------------------------------------------------
MAC Address     RADIUS Server  Authenticated  Time          Age  Dot1x
--------------------------------------------------------------------------------
0000.000074.3181  64.12.12.5     Yes            00d01h03m17s  Ena  Ena
```

The following table describes the information displayed by the **show auth-mac-addresses detailed** command.

TABLE 63 Output from the **show auth-mac-addresses detailed** command

Field	Description
Port	The port to which this information applies.
Dynamic-Vlan Assignment	Whether RADIUS dynamic VLAN assignment has been enabled for the port.
RADIUS failure action	What happens to traffic from a MAC address for which RADIUS authentication has failed either block the traffic or assign the MAC address to a restricted VLAN.
Failure restrict use dot1x	Indicates if 802.1x traffic that failed multi-device port authentication, but succeeded 802.1x authentication to gain access to the network.
Override-restrict-vlan	Whether a port can be dynamically assigned to a VLAN specified by a RADIUS server, if the port had been previously placed in the restricted VLAN because a previous attempt at authenticating a MAC address on that port failed.
Port Default Vlan	The VLAN to which the port is assigned, and whether the port had been dynamically assigned to the VLAN by a RADIUS server.
Port VLAN state	Indicates the state of the port VLAN. The State can be one of the following “Default”, “RADIUS Assigned” or “Restricted”.

TABLE 63 Output from the **show auth-mac-addresses detailed** command (Continued)

Field	Description
802.1X override Dynamic PVID	Indicates if 802.1X can dynamically assign a Port VLAN ID (PVID).
override return to PVID	If a port PVID is assigned through the multi-device port authentication feature, and 802.1X authentication subsequently specifies a different PVID, then the PVID specified through 802.1X authentication overrides the PVID specified through multi-device port authentication. This line indicates the PVID the port will use if 802.1X dynamically assigns PVID.
Original PVID	The originally configured (not dynamically assigned) PVID for the port.
DOS attack protection	Whether denial of service attack protection has been enabled for multi-device port authentication, limiting the rate of authentication attempts sent to the RADIUS server.
Accepted Mac Addresses	The number of MAC addresses that have been successfully authenticated.
Rejected Mac Addresses	The number of MAC addresses for which authentication has failed.
Authentication in progress	The number of MAC addresses for which authentication is pending. This is the number of MAC addresses for which an Access-Request message has been sent to the RADIUS server, and for which the RADIUS server has not yet sent an Access-Accept message.
Authentication attempts	The total number of authentication attempts made for MAC addresses on an interface, including pending authentication attempts.
RADIUS timeouts	The number of times the session between the Brocade device and the RADIUS server timed out.
RADIUS timeout action	Action to be taken by the RADIUS server if it times out.
MAC address on the PVID	Number of MAC addresses on the PVID.
MAC address authorized on PVID	Number of authorized MAC addresses on the PVID.
Aging of MAC-sessions	Whether software aging of MAC addresses is enabled.
Port move-back VLAN	Indicates the destination VLAN when a RADIUS assigned VLAN is removed. By default, it would return the configured VLAN.
Max-Age of sw MAC-sessions	The configured software aging period for MAC addresses.
hw age for denied MAC	The hardware aging period for blocked MAC addresses. The MAC addresses are dropped in hardware ones the aging period expires.
MAC Filter applied	Indicates whether a MAC address filter has been applied to this port to specify pre-authenticated MAC addresses.
Dynamic ACL applied	Indicates whether a dynamic ACL was applied to this port.
num Dynamic Tagged Vlan	The number of dynamically tagged VLANs on this port.
Dynamic Tagged Vlan list	The list of dynamically tagged VLANs on this port. In this example, **1025 (1/1)** indicates that there was one MAC session and one learned MAC address for VLAN 1025. Likewise, **4060 (1/0)** indicates that there was one MAC session and no learned MAC addresses for VLAN 4060.
MAC Address	The MAC addresses learned on the port. If the packet for which multi-device port authentication was performed also contained an IP address, then the IP address is displayed as well.
RADIUS Server	The IP address of the RADIUS server used for authenticating the MAC addresses.

TABLE 63 Output from the **show auth-mac-addresses detailed** command (Continued)

Field	Description
Authenticated	Whether the MAC address has been authenticated by the RADIUS server.
Time	The time at which the MAC address was authenticated. If the clock is set on the Brocade device, then the actual date and time are displayed. If the clock has not been set, then the time is displayed relative to when the device was last restarted.
Age	The age of the MAC address entry in the authenticated MAC address list.
Dot1x	Indicated if 802.1X authentication is enabled or disabled for the MAC address

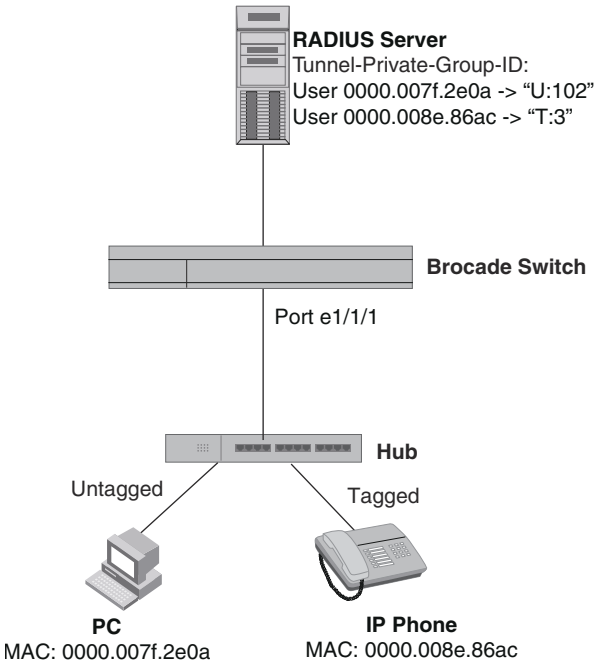
Example port authentication configurations

This section includes configuration examples of multi-device port authentication with dynamic VLAN assignment, and multi-device port authentication and 802.1X authentication.

Multi-device port authentication with dynamic VLAN assignment

Figure 11 illustrates multi-device port authentication with dynamic VLAN assignment on a Brocade device. In this configuration, a PC and an IP phone are connected to a hub, which is connected to port e1 on a Brocade device. The profile for the PC MAC address on the RADIUS server specifies that the PC should be dynamically assigned to VLAN 102, and the RADIUS profile for the IP phone specifies that it should be dynamically assigned to VLAN 3.

FIGURE 10 Using multi-device port authentication with dynamic VLAN assignment



In this example, multi-device port authentication is performed for both devices. If the PC is successfully authenticated, port e1 PVID is changed from VLAN 1 (the DEFAULT-VLAN) to VLAN 102. If authentication for the PC fails, then the PC can be placed in a specified "restricted" VLAN, or traffic from the PC can be blocked in hardware. In this example, if authentication for the PC fails, the PC would be placed in VLAN 1023, the restricted VLAN.

If authentication for the IP phone is successful, then port e1/1/1 is added to VLAN 3. If authentication for the IP phone fails, then traffic from the IP phone would be blocked in hardware. (Devices sending tagged traffic cannot be placed in the restricted VLAN.)

The portion of the running-config related to multi-device port authentication is as follows.

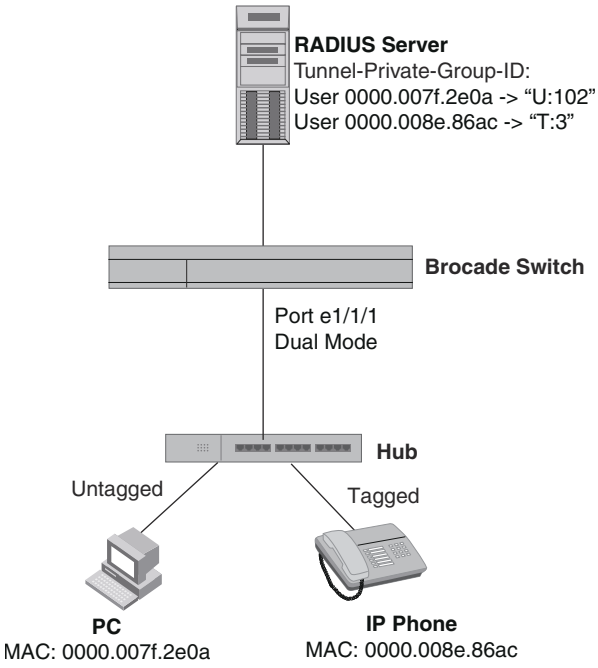
```
mac-authentication enable
mac-authentication auth-fail-vlan-id 1023

interface ethernet 1
 dual-mode
 mac-authentication enable
 mac-authentication auth-fail-action restrict-vlan
 mac-authentication enable-dynamic-vlan
 mac-authentication disable-ingress-filtering
```

The **mac-authentication disable-ingress-filtering** command enables tagged packets on the port, even if the port is not a member of the VLAN. If this feature is not enabled, authentication works as in "Example 1— Multi-device port authentication with dynamic VLAN assignment"

Example 1— Multi-device port authentication with dynamic VLAN assignment

Figure 11 illustrates multi-device port authentication with dynamic VLAN assignment on a Brocade device. In this configuration, a PC and an IP phone are connected to a hub, which is connected to port e1/1/1 on a Brocade device. Port e1 is configured as a dual-mode port. Also, **mac-authentication disable-ingress-filtering** is enabled on the port. The profile for the PC MAC address on the RADIUS server specifies that the PC should be dynamically assigned to VLAN 102, and the RADIUS profile for the IP phone specifies that it should be dynamically assigned to VLAN 3.

FIGURE 11 Using multi-device port authentication with dynamic VLAN assignment



In this example, multi-device port authentication is performed for both devices. If the PC is successfully authenticated, dual-mode port e1 PVID is changed from the VLAN 1 (the DEFAULT-VLAN) to VLAN 102. If authentication for the PC fails, then the PC can be placed in a specified "restricted" VLAN, or traffic from the PC can be blocked in hardware. In this example, if authentication for the PC fails, the PC would be placed in VLAN 1023, the restricted VLAN.

If authentication for the IP phone is successful, then dual-mode port e1/1/1 is added to VLAN 3. If authentication for the IP phone fails, then traffic from the IP phone would be blocked in hardware. (Devices sending tagged traffic cannot be placed in the restricted VLAN.)

NOTE
This example assumes that the IP phone initially transmits untagged packets (for example, CDP or DHCP packets), which trigger the authentication process on the Brocade device and client lookup on the RADIUS server. If the phone sends only tagged packets and the port (e1/1/1) is not a member of that VLAN, authentication would not occur. In this case, port e1 must be added to that VLAN prior to authentication.

The part of the running-config related to multi-device port authentication would be as follows.

```
mac-authentication enable
mac-authentication auth-fail-vlan-id 1023

interface ethernet 1/1/1
 mac-authentication enable
 mac-authentication auth-fail-action restrict-vlan
 mac-authentication enable-dynamic-vlan
 dual-mode
```

Examples of multi-device port authentication and 802.1X authentication configuration on the same port

The following examples show configurations that use multi-device port authentication and 802.1X authentication on the same port.

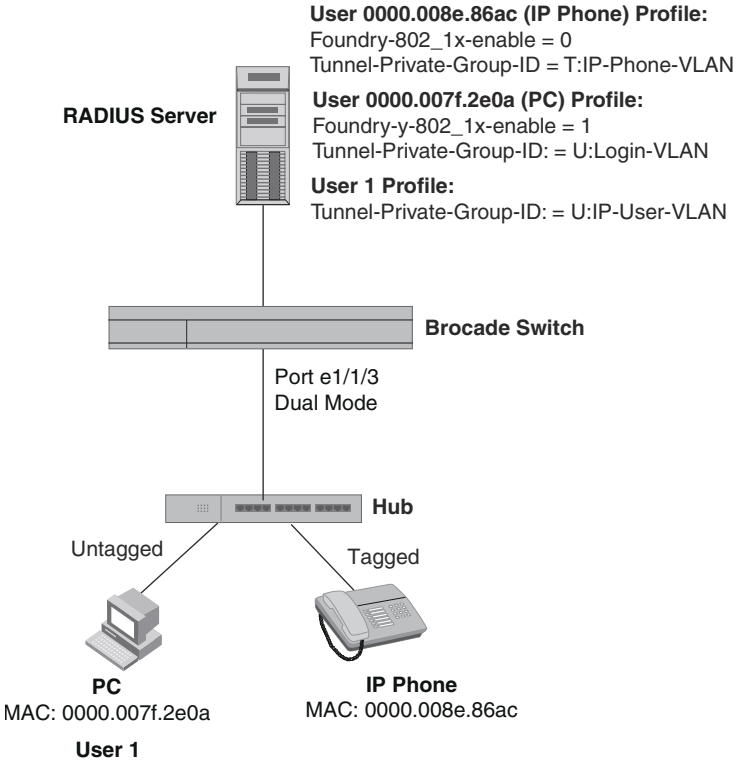
Example 1 — Multi-device port authentication and 802.1x authentication on the same port

Figure 12 illustrates an example configuration that uses multi-device port authentication and 802.1X authentication n the same port. In this configuration, a PC and an IP phone are connected to port e 1/1/3 on a Brocade device. Port e 1/1/3 is configured as a dual-mode port.

The profile for the PC MAC address on the RADIUS server specifies that the PC should be dynamically assigned to VLAN "Login-VLAN", and the RADIUS profile for the IP phone specifies that it should be dynamically assigned to the VLAN named "IP-Phone-VLAN". When User 1 is successfully authenticated using 802.1X authentication, the PC is then placed in the VLAN named "User-VLAN".

NOTE
This example assumes that the IP phone initially transmits untagged packets (for example, CDP or DHCP packets), which trigger the authentication process on the Brocade device and client lookup on the RADIUS server. If the phone sends only tagged packets and the port (e 1/1/3) is not a member of that VLAN, authentication would not occur. In this case, port e 1/1/3 must be added to that VLAN prior to authentication.

FIGURE 12 Using multi-device port authentication and 802.1X authentication on the same port



When the devices attempt to connect to the network, they are first subject to multi-device port authentication.

When the MAC address of the IP phone is authenticated, the Access-Accept message from the RADIUS server specifies that the IP phone port be placed into the VLAN named “IP-Phone-VLAN”. which is VLAN 7. The Foundry-802_1x-enable attribute is set to 0, meaning that 802.1X authentication is skipped for this MAC address. Port e 1/1/3 is placed in VLAN 7 as a tagged port. No further authentication is performed.

When the PC MAC address is authenticated, the Access-Accept message from the RADIUS server specifies that the PVID for the PC port be changed to the VLAN named “Login-VLAN”, which is VLAN 1024. The Foundry-802_1x-enable attribute is set to 1, meaning that 802.1X authentication is required for this MAC address. The PVID of the port e 1/1/3 is temporarily changed to VLAN 1024, pending 802.1X authentication.

When User 1 attempts to connect to the network from the PC, he is subject to 802.1X authentication. If User 1 is successfully authenticated, the Access-Accept message from the RADIUS server specifies that the PVID for User 1 port be changed to the VLAN named “User-VLAN”, which is VLAN 3. If 802.1X authentication for User 1 is unsuccessful, the PVID for port e 1/1/3 is changed to that of the restricted VLAN, which is 1023, or untagged traffic from port e 1/1/3 can be blocked in hardware.

The part of the running-config related to port e 1/1/3 would be as follows.

```
interface ethernet 1/1/3
 dot1x port-control auto
 mac-authentication enable
 dual-mode
```

When the PC is authenticated using multi-device port authentication, the port PVID is changed to “Login-VLAN”, which is VLAN 1024 in this example.

When User 1 is authenticated using 802.1X authentication, the port PVID is changed to “User-VLAN”, which is VLAN 3 in this example.

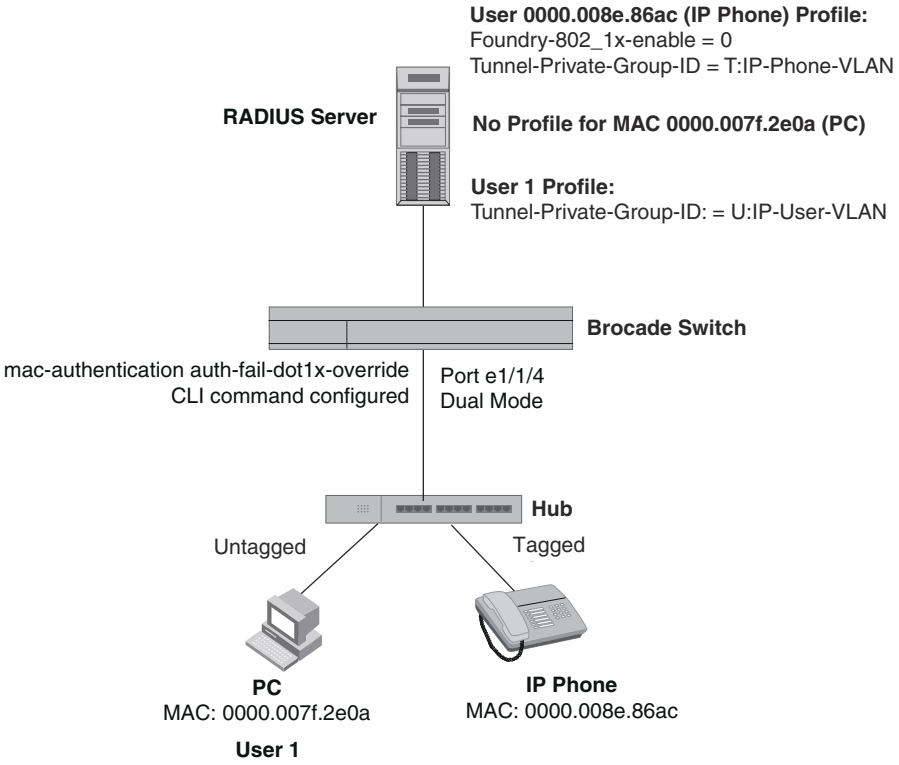
Example 2 — Creating a profile on the RADIUS server for each MAC address

The configuration in Figure 13 requires that you create a profile on the RADIUS server for each MAC address to which a device or user can connect to the network. In a large network, this can be difficult to implement and maintain.

As an alternative, you can create MAC address profiles only for those devices that do not support 802.1X authentication, such as IP phones and printers, and configure the device to perform 802.1X authentication for the other devices that do not have MAC address profiles, such as user PCs. To do this, you configure the device to perform 802.1X authentication when a device fails multi-device port authentication.

Figure 13 shows a configuration where multi-device port authentication is performed for an IP phone, and 802.1X authentication is performed for a user PC. There is a profile on the RADIUS server for the IP phone MAC address, but not for the PC MAC address.

FIGURE 13 802.1X Authentication is performed when a device fails multi-device port authentication



Multi-device port authentication is initially performed for both devices. The IP phone MAC address has a profile on the RADIUS server. This profile indicates that 802.1X authentication should be skipped for this device, and that the device port be placed into the VLAN named “IP-Phone-VLAN”.

Since there is no profile for the PC MAC address on the RADIUS server, multi-device port authentication for this MAC address fails. Ordinarily, this would mean that the PVID for the port would be changed to that of the restricted VLAN, or traffic from this MAC would be blocked in hardware. However, the device is configured to perform 802.1X authentication when a device fails multi-device port authentication, so when User 1 attempts to connect to the network from the PC, he is subject to 802.1X authentication. If User 1 is successfully authenticated, the PVID for port e 1/1/4 is changed to the VLAN named “User-VLAN”.

NOTE
This example assumes that the IP phone initially transmits untagged packets (for example, CDP or DHCP packets), which trigger the authentication process on the Brocade device and client lookup on the RADIUS server. If the phone sends only tagged packets and the port (e 1/1/4) is not a member of that VLAN, authentication would not occur. In this case, port e 1/1/4 must be added to that VLAN prior to authentication.

To configure the device to perform 802.1X authentication when a device fails multi-device port authentication, enter the following command.

```
Brocade(config)# mac-authentication auth-fail-dot1x-override
```

Syntax: [no] mac-authentication auth-fail-dot1x-override

Chapter

10

DoS Attack Protection

Table 64 lists DoS protection features supported in Brocade ICX 6650. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where noted.

TABLE 64 Supported DoS protection features

Feature	Brocade ICX 6650
Smurf attack (ICMP attack) protection	Yes
TCP SYN attack protection	Yes

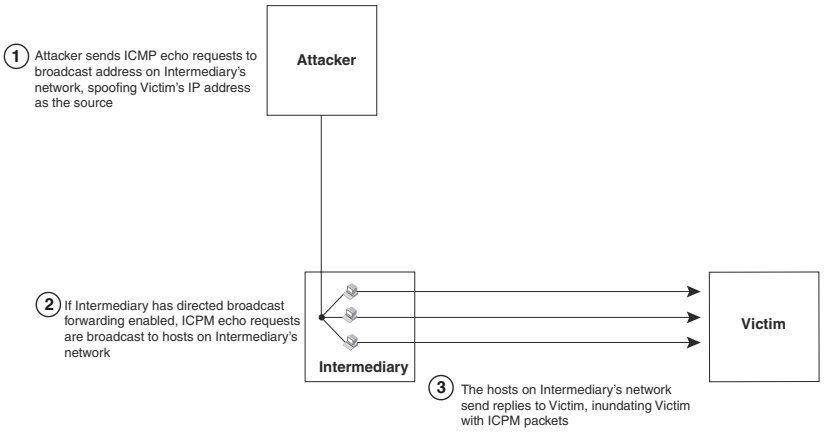
This chapter explains how to protect your Brocade devices from Denial of Service (DoS) attacks.

In a Denial of Service (DoS) attack, a device is flooded with useless packets, hindering normal operation.Brocade devices include measures for defending against two types of DoS attacks Smurf attacks and TCP SYN attacks.

Smurf attacks

A *Smurf attack* is a kind of DoS attack in which an attacker causes a victim to be flooded with Internet Control Message Protocol (ICMP) echo (Ping) replies sent from another network. Figure 14 illustrates how a Smurf attack works.

FIGURE 14 How a Smurf attack floods a victim with ICMP replies



The attacker sends an ICMP echo request packet to the broadcast address of an intermediary network. The ICMP echo request packet contains the spoofed address of a victim network as its source. When the ICMP echo request reaches the intermediary network, it is converted to a Layer 2 broadcast and sent to the hosts on the intermediary network. The hosts on the intermediary network then send ICMP replies to the victim network.

For each ICMP echo request packet sent by the attacker, a number of ICMP replies equal to the number of hosts on the intermediary network are sent to the victim. If the attacker generates a large volume of ICMP echo request packets, and the intermediary network contains a large number of hosts, the victim can be overwhelmed with ICMP replies.

Avoiding being an intermediary in a Smurf attack

A Smurf attack relies on the intermediary to broadcast ICMP echo request packets to hosts on a target subnet. When the ICMP echo request packet arrives at the target subnet, it is converted to a Layer 2 broadcast and sent to the connected hosts. This conversion takes place only when directed broadcast forwarding is enabled on the device.

To avoid being an intermediary in a Smurf attack, make sure forwarding of directed broadcasts is disabled on the Brocade device. Directed broadcast forwarding is disabled by default. To disable directed broadcast forwarding, do one of the following.

```
Brocade(config)# no ip directed-broadcast
```

Syntax: [no] **ip directed-broadcast**

Avoiding being a victim in a Smurf attack

You can configure the Brocade device to drop ICMP packets when excessive numbers are encountered, as is the case when the device is the victim of a Smurf attack. You can set threshold values for ICMP packets that are targeted at the router itself or passing through an interface, and drop them when the thresholds are exceeded.

For example, to set threshold values for ICMP packets targeted at the router, enter the following command in global CONFIG mode.

```
Brocade(config)# ip icmp burst-normal 5000 burst-max 10000 lockup 300
```

To set threshold values for ICMP packets received on interface 1/1/3, enter the following commands.

```
Brocade(config)# interface ethernet 1/1/3
Brocade(config-if-e10000-1/1/3)# ip icmp burst-normal 5000 burst-max 10000 lockup
300
```

For Layer 3 router code, if the interface is part of a VLAN that has a router VE, you must configure ICMP attack protection at the VE level. Otherwise, you can configure this feature at the interface level as shown in the previous example. When ICMP attack protection is configured at the VE level, it will apply to routed traffic only. It will not affect switched traffic.

NOTE
You must configure VLAN information for the port *before* configuring ICMP attack protection. You cannot change the VLAN configuration for a port on which ICMP attack protection is enabled.

To set threshold values for ICMP packets received on VE 31, enter commands such as the following.

```
Brocade(config)# interface ve 31
Brocade(config-vif-31)# ip icmp burst-normal 5000 burst-max 10000 lockup 300
```

Syntax: **ip icmp burst-normal** *value* **burst-max** *value* **lockup** *seconds*

The **burst-normal** value parameter can be from 1 through 100,000 packets per second.

The **burst-max** value parameter can be from 1 through 100,000 packets per second.

The **lockup** value parameter can be from 1 through 10,000 seconds.

This command is supported on Ethernet and Layer 3 interfaces.

The number of incoming ICMP packets per second is measured and compared to the threshold values as follows:

- If the number of ICMP packets exceeds the **burst-normal** value, the excess ICMP packets are dropped.
- If the number of ICMP packets exceeds the **burst-max** value, all ICMP packets are dropped for the number of seconds specified by the **lockup** value. When the lockup period expires, the packet counter is reset and measurement is restarted.

In the example, if the number of ICMP packets received per second exceeds 5,000, the excess packets are dropped. If the number of ICMP packets received per second exceeds 10,000, the device drops all ICMP packets for the next 300 seconds (5 minutes).

TCP SYN attacks

TCP SYN attacks exploit the process of how TCP connections are established to disrupt normal traffic flow. When a TCP connection starts, the connecting host first sends a TCP SYN packet to the destination host. The destination host responds with a SYN ACK packet, and the connecting host sends back an ACK packet. This process, known as a "TCP three-way handshake," establishes the TCP connection.

While waiting for the connecting host to send an ACK packet, the destination host keeps track of the as-yet incomplete TCP connection in a connection queue. When the ACK packet is received, information about the connection is removed from the connection queue. Usually there is not much time between the destination host sending a SYN ACK packet and the source host sending an ACK packet, so the connection queue clears quickly.

In a TCP SYN attack, an attacker floods a host with TCP SYN packets that have random source IP addresses. For each of these TCP SYN packets, the destination host responds with a SYN ACK packet and adds information to the connection queue. However, because the source host does not exist, no ACK packet is sent back to the destination host, and an entry remains in the connection queue until it ages out (after approximately a minute). If the attacker sends enough TCP SYN packets, the connection queue can fill up, and service can be denied to legitimate TCP connections.

To protect against TCP SYN attacks, you can configure the Brocade device to drop TCP SYN packets when excessive numbers are encountered. You can set threshold values for TCP SYN packets that are targeted at the router itself or passing through an interface, and drop them when the thresholds are exceeded.

For example, to set threshold values for TCP SYN packets targeted at the router, enter the following command in global CONFIG mode.

```
Brocade(config)# ip tcp burst-normal 10 burst-max 100 lockup 300
```

To set threshold values for TCP SYN packets received on interface 1/1/3, enter the following commands.

```
Brocade(config)# interface ethernet 1/1/3
Brocade(config-if-e10000-1/1/3)# ip tcp burst-normal 10 burst-max 100 lockup 300
```

For Layer 3 router code, if the interface is part of a VLAN that has a router VE, you must configure TCP/SYN attack protection at the VE level. Otherwise, you can configure this feature at the interface level as shown in the previous example. When TCP/SYN attack protection is configured at the VE level, it will apply to routed traffic only. It will not affect switched traffic.

NOTE
You must configure VLAN information for the port *before* configuring TCP/SYN attack protection. You cannot change the VLAN configuration for a port on which TCP/SYN attack protection is enabled.

To set threshold values for TCP/SYN packets received on VE 31, enter commands such as the following.

```
Brocade(config)# interface ve 31
Brocade(config-vif-31)# ip tcp burst-normal 5000 burst-max 10000 lockup 300
```

Syntax: **ip tcp burst-normal** *value* **burst-max** *value* **lockup** *seconds*

NOTE
This command is available at the global CONFIG level on both Chassis devices and Compact devices. On Chassis devices, this command is available at the Interface level as well. This command is supported on Ethernet and Layer 3 interfaces.

The **burst-normal** *value* parameter can be from 1 – 100,000 packets per second.

The **burst-max** *value* parameter can be from 1 – 100,000 packets per second.

The **lockup** *value* parameter can be from 1 – 10,000 seconds.

The number of incoming TCP SYN packets per second is measured and compared to the threshold values as follows:

- If the number of TCP SYN packets exceeds the **burst-normal** value, the excess TCP SYN packets are dropped.
- If the number of TCP SYN packets exceeds the **burst-max** value, all TCP SYN packets are dropped for the number of seconds specified by the **lockup** value. When the lockup period expires, the packet counter is reset and measurement is restarted.

In the example, if the number of TCP SYN packets received per second exceeds 10, the excess packets are dropped. If the number of TCP SYN packets received per second exceeds 100, the device drops all TCP SYN packets for the next 300 seconds (5 minutes).

TCP security enhancement

TCP security enhancement improves upon the handling of TCP inbound segments. This enhancement eliminates or minimizes the possibility of a TCP reset attack, in which a perpetrator attempts to prematurely terminate an active TCP session, and a data injection attack, wherein an attacker injects or manipulates data in a TCP connection.

In both cases, the attack is blind, meaning the perpetrator does not have visibility into the content of the data stream between two devices, but blindly injects traffic. Also, the attacker does not see the direct effect, the continuing communications between the devices and the impact of the injected packet, but may see the indirect impact of a terminated or corrupted session.

The TCP security enhancement prevents and protects against the following three types of attacks:

- Blind TCP reset attack using the reset (RST) bit
- Blind TCP reset attack using the synchronization (SYN) bit
- Blind TCP packet injection attack

The TCP security enhancement is automatically enabled.

Protecting against a blind TCP reset attack using the RST bit

In a blind TCP reset attack using the RST bit, a perpetrator attempts to guess the RST bits to prematurely terminate an active TCP session.

To prevent a user from using the RST bit to reset a TCP connection, the RST bit is subject to the following rules when receiving TCP segments:

- If the RST bit is set and the sequence number is outside the expected window, the Brocade device silently drops the segment.
- If the RST bit is exactly the next expected sequence number, the Brocade device resets the connection.
- If the RST bit is set and the sequence number does not exactly match the next expected sequence value, but is within the acceptable window, the Brocade device sends an acknowledgement.

Protecting against a blind TCP reset attack using the SYN bit

In a blind TCP reset attack using the SYN bit, a perpetrator attempts to guess the SYN bits to prematurely terminate an active TCP session.

To prevent a user from using the SYN bit to tear down a TCP connection, in current software releases, the SYN bit is subject to the following rules when receiving TCP segments:

- If the SYN bit is set and the sequence number is outside the expected window, the Brocade device sends an acknowledgement (ACK) back to the peer.
- If the SYN bit is set and the sequence number is an exact match to the next expected sequence, the Brocade device sends an ACK segment to the peer. Before sending the ACK segment, the software subtracts one from the value being acknowledged.
- If the SYN bit is set and the sequence number is acceptable, the Brocade device sends an acknowledgement (ACK) segment to the peer.

Protecting against a blind injection attack

In a blind TCP injection attack, a perpetrator tries to inject or manipulate data in a TCP connection.

To reduce the chances of a blind injection attack, an additional check on all incoming TCP segments is performed.

Displaying statistics about packets dropped because of DoS attacks

To display information about ICMP and TCP SYN packets dropped because burst thresholds were exceeded, enter the **show statistics dos-attack** command.

```
Brocade# show statistics dos-attack
--------------------------- Local Attack Statistics --------------------------
ICMP Drop Count    ICMP Block Count    SYN Drop Count    SYN Block Count
---------------    ----------------    --------------    ---------------
              0                   0                 0                   0
--------------------------- Transit Attack Statistics -------------------------
Port   ICMP Drop Count    ICMP Block Count    SYN Drop Count    SYN Block Count
-----  ---------------    ----------------    --------------    ---------------
1/1/1       0                   0                 0                   0
```

Syntax: show statistics dos-attack

To clear statistics about ICMP and TCP SYN packets dropped because burst thresholds were exceeded, enter the **clear statistics dos-attack** command.

```
Brocade# clear statistics dos-attack
```

Syntax: clear statistics dos-attack

Chapter

11

Rate Limiting and Rate Shaping

Table 65 lists the rate limiting and rate shaping features supported on Brocade ICX 6650. These features are supported in the Layer 2, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 65 Supported rate limiting and rate shaping features

Feature	Brocade ICX 6650
Inbound rate limiting (Port-based rate limiting on inbound ports)	Yes
Outbound rate shaping	Yes

This chapter describes how to implement and configure port-based rate limiting and rate shaping on Brocade ICX 6650.

Port-based rate limiting

Rate limiting applies to inbound ports and rate shaping applies to outbound ports. Port-based fixed rate limiting is supported on inbound ports. This feature allows you to specify the maximum number of *packets* a given port can receive. The port drops packets that exceed the limit you specify. You can configure a Fixed rate limiting policy on a port inbound direction only. Fixed rate limiting applies to all traffic on the rate limited port.

Fixed rate limiting is at line rate and occurs in hardware. Refer to "Rate limiting in hardware" on page 274.

The Fixed rate limiting policy applies to one-second intervals and allows the port to receive the number of packets you specify in the policy, but drops additional packets. Unused bandwidth is not carried over from one interval to the next.

NOTE
Port based Rate Limiting affects only known-unicast traffic. Broadcast, Multicast and Unknown-unicast (BUM) is not affected by this rate. To rate limit the BUM traffic, use BUM rate limiting as described in chapter BUM Rate Limiting.

NOTE
Brocade recommends that you do not use Fixed rate limiting on ports that receive route control traffic or Spanning Tree Protocol (STP) control traffic. If the port drops control packets due to the Fixed rate limiting policy, routing or STP can be disrupted.

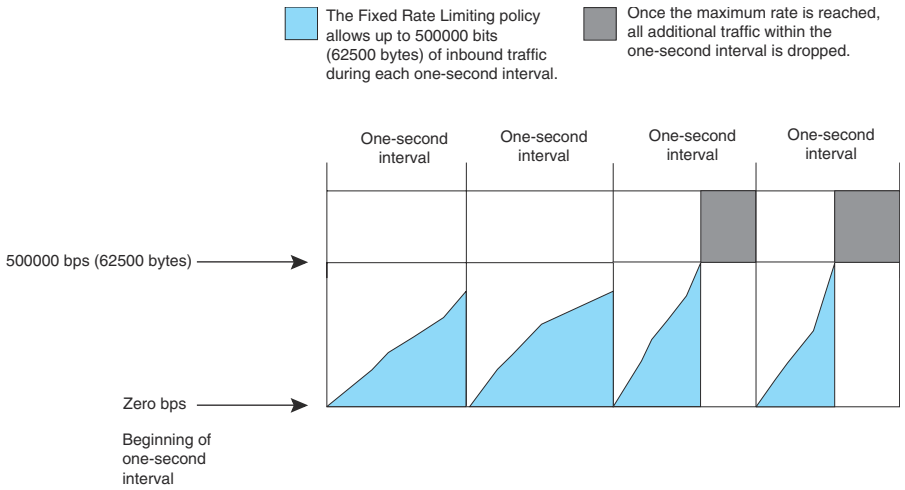
How port-based fixed rate limiting works

Fixed rate limiting counts the number of packets that a port receives, in one second intervals. If the number exceeds the maximum number you specify when you configure the rate, the port drops all further inbound packets for the duration of the one-second interval.

After the one-second interval is complete, the port clears the counter and re-enables traffic.

Figure 15 shows an example of how Fixed rate limiting works. In this example, a Fixed rate limiting policy is applied to a port to limit the inbound traffic to 500000 packets a second. During the first two one-second intervals, the port receives less than 500000 packets in each interval. However, the port receives more than 500000 packets during the third and fourth one-second intervals, and consequently drops the excess traffic.

FIGURE 15 Fixed rate limiting



NOTE

The software counts the packets by polling statistics counters for the port every 100 milliseconds, which provides 10 readings each second. Due to the polling interval, the Fixed Rate Limiting policy has an accuracy of within 10% of the port's line rate. It is therefore possible for the policy to sometimes allow more traffic than the limit you specify, but the extra traffic is never more than 10% of the port's line rate.

Rate limiting in hardware

Each Brocade device supports in hardware rate limiting at line-rate. The device creates entries in Content Addressable Memory (CAM) for the rate limiting policies. The CAM entries enable the device to perform the rate limiting in hardware instead of sending the traffic to the CPU. The device sends the first packet in a given traffic flow to the CPU, which creates a CAM entry for the traffic flow. A CAM entry consists of the source and destination addresses of the traffic. The device uses the CAM entry for rate limiting all the traffic within the same flow. A rate limiting CAM entry remains in the CAM for two minutes before aging out.

Configuration notes for port-based fixed rate limiting

- Rate limiting is available only on inbound ports.
- The rate limit on IPv6 hardware takes several seconds to take effect at higher configured rate limit values. For example, if the configured rate limit is 1500000 packets/second, line-rate limiting could take up to 43 seconds to take effect.

Configuring a port-based fixed rate limiting policy

To configure rate limiting on a port, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/2/4
Brocade(config-if-e10000-1/2/4)# rate input fixed 500
```

These commands configure a fixed rate limiting policy that allows port 24 to receive a maximum of 500 packets per second. If the port receives additional packets during a given one-second interval, the port drops all inbound packets on the port until the next one-second interval starts.

Syntax: [no] **rate-limit input fixed** *average-rate*

For Brocade ICX 6650 devices, the *average-rate* parameter specifies the maximum number of *packets* per second (pkts/s) the port can receive. The minimum rate that can be configured is 125 pkts/s.

Displaying the port-based fixed rate limiting configuration

To display the fixed rate limiting configuration on the device, use the **show rate-limit input** command as shown below.

```
Brocade# show rate-limit input
Total rate-limited interface count: 5.
    Port     Configured Input Rate        Actual Input Rate
   1/1/1            65000 pkts/sec           65000 pkts/sec
   1/1/2            95000 pkts/sec          195000 pkts/sec
   1/1/6             1950 pkts/sec            1950 pkts/sec
   1/2/2           230432 pkts/sec          230000 pkts/sec
   1/2/6           234113 pkts/sec          234000 pkts/sec
```

Syntax: show rate-limit input

This command lists the ports on which fixed rate limiting is configured, and provides the information listed in Table 66 for each of the ports.

TABLE 66 CLI display of Fixed rate limiting information

Field	Description
Total rate-limited interface count	The total number of ports that are configured for Fixed rate limiting.
Port	The port number.
Configured Input Rate	The maximum rate requested for inbound traffic. The rate is measured in packets per second (pkts/s).
Actual Input Rate	The actual maximum rate provided by the hardware. The rate is measured in packets per second (pkts/s).

Rate shaping

Outbound Rate Shaping is a port-level feature for shaping the rate and controlling the bandwidth of outbound traffic on a port. This feature smooths out excess and bursty traffic to the configured maximum limit before it is sent out on a port. Packets are stored in available buffers and then forwarded at a rate no greater than the configured limit. This process provides for better control over the inbound traffic of neighboring devices.

The device has one global rate shaper for a port and one rate shaper for each port priority queue. Rate shaping is done on a single-token basis, where each token is defined to be 1 packet.

Configuration notes for rate shaping

The following rules apply when configuring outbound rate shapers:

- Outbound rate shapers can be configured *only* on physical ports, not on virtual or loopback ports.
- For trunk ports, the rate shaper must be configured on individual ports of a trunk using the **config-trunk-ind** command (trunk configuration level); you cannot configure a rate shaper for a trunk.
- When outbound rate shaping is enabled on a port on an IPv4 device, the port QoS queuing method (**qos mechanism**) will be strict mode. This applies to IPv4 devices only. On IPv6 devices, the QoS mechanism is whatever method is configured on the port, even when outbound rate shaping is enabled.
- You can configure a rate shaper for a port and for the individual priority queues of that port. However, if a port rate shaper is configured, that value overrides the rate shaper value of a priority queue if the priority queue rate shaper is greater than the rate shaper for the port.

The configured rate shaper values are rounded up to the nearest multiples of minimum values supported on the platform. Table 67 shows the minimum and the maximum values for output rate shaping on Brocade ICX 6650.

TABLE 67 Output rate shaping on Brocade ICX 6650 devices

Device	Module	Minimum	Maximum
Brocade ICX 6650	40 Gbps ports	20 pkts/sec	80000000 pkts/sec
Brocade ICX 6650	10 Gbps ports	20 pkts/sec	20000000 pkts/sec

Configuring outbound rate shaping for a port

To configure the maximum rate at which outbound traffic is sent out on a port, enter the following commands.

```
Brocade(config)#interface ethernet 1/1/2
Brocade(config-if-e10000-1/1/2)#rate-limit output shaping 1300
```

Syntax: [**no**] **rate-limit output shaping** *value*

Configuring outbound rate shaping for a specific priority

To configure the maximum rate at which outbound traffic is sent out on a port priority queue, enter the following commands.

```
Brocade(config)#interface ethernet 1/1/2
Brocade(config-if-e10000-1/1/2)#rate-limit output shaping 500 priority 7
```

Syntax: [**no**] **rate-limit output shaping** *value* **priority** *priority-queue*

Specify 0-7 for *priority-queue*

Configuring outbound rate shaping for a trunk port

This feature is supported on individual ports of a static trunk group and on LACP trunk ports. To configure the maximum rate at which outbound traffic is sent out on a trunk port, enter the following on each trunk port where outbound traffic will be shaped.

```
Brocade(config)# trunk ethernet 1/1/3 to 1/1/6
Brocade(config-trunk-3-6)# config-trunk-ind
Brocade(config-trunk-3-6)# rate-limit output shaping ethernet 1/1/5 651
Brocade(config-trunk-3-6)# rate-limit output shaping ethernet 1/1/4 1300
```

The above commands configure an outbound rate shaper on port 1/1/4 and port 1/1/5.

Syntax: [**no**] **rate-limit output shaping ethernet** *port value*

Specify the *port* variable in *stack-unit/slotnum/portnum* format.

Displaying rate shaping configurations

To display the configured outbound rate shaper on a device, use the **show rate-limit output-shaping** command.

```
Brocade# show rate-limit output-shaping
Outbound Rate Shaping Limits in Packets/sec:
 Port   PortMax   Prio0   Prio1   Prio2   Prio3   Prio4   Prio5   Prio6   Prio7
 1/1/1      -       -       -       -       -       -       -       -     651
 1/1/2      1302    -       -       -       -       -       -       -       -
 1/1/5      651     -       -       -       -       -       -       -       -
```

The output lists the ports on a device, the configured outbound rate shaper (if any) for each port, and the priority for a port.

CPU rate-limiting

Unnecessary traffic to the switch CPU lowers the efficiency of the CPU and delays handling of other traffic that requires processing. CPU rate limiting is a CPU protection scheme which limits certain traffic types.

CPU rate limiting identifies the traffic type and assigns a maximum rate limit to the traffic type. The traffic types which are subjected to rate limiting include broadcast ARP and other exceptions, such as TTL exceed, IP MTU failed, reverse path check failed, IP fragments, and unsupported tunneling. Each of these types is rate-limited individually.

Table 68 shows the rate limits for each rate-limited packet type and the platforms to which each rate limit applies. These rates cannot be configured by users currently.

TABLE 68 CPU rate limits for packet type and applicable platforms

Packet type	Rate limit in packets per second
ARP	6000
IP TTL exceed, or Reverse path check failed	150
IP MTU exceed, IP tunnel-terminated packets which are fragmented or has options, or IP tunnel-terminated packets with unsupported GRE tunnel header	3000
IP Unicast packets mirrored to CPU due to ICMP redirect	100
Bridge packets forward to CPU	5000

Chapter

DHCP 12

Table 69 lists the Dynamic Host Configuration Protocol (DHCP) packet inspection and tracking features supported in Brocade ICX 6650. These features are supported in the Layer 2, base Layer 3, edge Layer 3, and full Layer 3 software images, except where explicitly noted.

TABLE 69 Supported DHCP packet inspection and tracking features

Feature	Brocade ICX 6650
Dynamic ARP inspection	Yes
DHCP snooping	Yes
DHCP relay agent information (DHCP Option 82)	Yes
IP source guard	Yes

Dynamic ARP inspection

For enhanced network security, you can configure the Brocade device to inspect and keep track of Dynamic Host Configuration Protocol (DHCP) assignments.

Dynamic ARP Inspection (DAI) enables the Brocade device to intercept and examine all ARP request and response packets in a subnet and discard those packets with invalid IP to MAC address bindings. DAI can prevent common man-in-the-middle (MiM) attacks such as ARP cache poisoning, and disallow mis-configuration of client IP addresses.

ARP poisoning

ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address. Before a host can talk to another host, it must map the IP address to a MAC address first. If the host does not have the mapping in its ARP table, it creates an ARP request to resolve the mapping. All computers on the subnet will receive and process the ARP requests, and the host whose IP address matches the IP address in the request will send an ARP reply.

An ARP poisoning attack can target hosts, switches, and routers connected to the Layer 2 network by poisoning the ARP caches of systems connected to the subnet and by intercepting traffic intended for other hosts on the subnet. For instance, a malicious host can reply to an ARP request with its own MAC address, thereby causing other hosts on the same subnet to store this information in their ARP tables or replace the existing ARP entry. Furthermore, a host can send gratuitous replies without having received any ARP requests. A malicious host can also send out ARP packets claiming to have an IP address that actually belongs to another host (e.g. the default router). After the attack, all traffic from the device under attack flows through the attacker computer and then to the router, switch, or host.

Dynamic ARP Inspection

Dynamic ARP Inspection (DAI) allows only valid ARP requests and responses to be forwarded.

A Brocade device on which DAI is configured does the following:

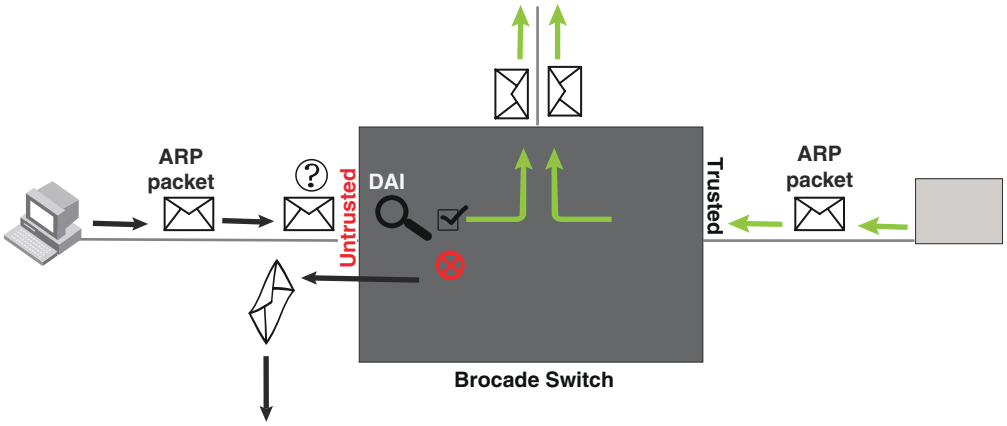
- Intercepts ARP packets received by the system CPU
- Inspects all ARP requests and responses received on untrusted ports
- Verifies that each of the intercepted packets has a valid IP-to-MAC address binding before updating the local ARP table, or before forwarding the packet to the appropriate destination
- Drops invalid ARP packets

When you enable DAI on a VLAN, by default, all member ports are untrusted. You must manually configure trusted ports. In a typical network configuration, ports connected to host ports are untrusted. You configure ports connected to other switches or routers as trusted.

DAI inspects ARP packets received on untrusted ports, as shown in Figure 16. DAI carries out the inspection based on IP-to-MAC address bindings stored in a trusted binding database. For the Brocade device, the binding database is the ARP table, which supports DAI, DHCP snooping, and IP Source Guard. To inspect an ARP request packet, DAI checks the source IP and source MAC address against the ARP table. For an ARP reply packet, DAI checks the source IP, source MAC, destination IP, and destination MAC addresses. DAI forwards the valid packets and discards those with invalid IP-to-MAC address bindings.

When ARP packets reach a trusted port, DAI lets them through, as shown in Figure 16.

FIGURE 16 Dynamic ARP inspection at work



ARP entries

DAI uses the IP/MAC mappings in the ARP table to validate ARP packets received on untrusted ports.

ARP entries in the ARP table derive from the following:

- **Dynamic ARP** – normal ARP learned from trusted ports.
- **Static ARP** – statically configured IP/MAC/port mapping.
- **Inspection ARP** – statically configured IP/MAC mapping, where the port is initially unspecified. The actual physical port mapping will be resolved and updated from validated ARP packets.Refer to "Configuring an inspection ARP entry" on page 282.

- **DHCP-Snooping ARP** – information collected from snooping DHCP packets when DHCP snooping is enabled on VLANs.

The status of an ARP entry is either pending or valid:

- **Valid** – the mapping is valid, and the port is resolved. This is always the case for static ARP entries.
- **Pending** – for normal dynamic and inspection ARP entries before they are resolved, and the port mapped. Their status changes to valid when they are resolved, and the port mapped.

Refer to also "System reboot and the binding database" on page 285.

Configuration notes and feature limitations for DAI

The following limits and restrictions apply when configuring DAI:

- To run Dynamic ARP Inspection, you must first enable support for ACL filtering based on VLAN membership or VE port membership. To do so, enter the following commands at the Global CONFIG Level of the CLI.

```
Brocade(config)# enable ACL-per-port-per-vlan
Brocade(config)# write memory
Brocade(config)# exit
Brocade# reload
```

NOTE
You must save the configuration and reload the software to place the change into effect.

- Brocade recommends that you do not enable DAI on a trunk port.
- The maximum number of DHCP and static DAI entries depends on the maximum number of ARP table entries allowed on the device. A Brocade ICX 6650 Layer 2 switch can have up to 4096 ARP entries and a Brocade ICX 6650 Layer 3 switch can have up to 64,000 ARP entries. In a Brocade ICX 6650 Layer 3 switch, you can use the **system-max ip-arp** command to change the maximum number of ARP entries for the device.

  However, only up to 1024 DHCP entries can be saved to flash.
- ACLs are supported on member ports of a VLAN on which DHCP snooping and Dynamic ARP Inspection (DAI) are enabled.
- DAI is supported on a VLAN without a VE, or on a VE with or without an assigned IP address.

Dynamic ARP inspection configuration

Configuring DAI consists of the following steps.

1. Configure inspection ARP entries for hosts on untrusted ports.Refer to "Configuring an inspection ARP entry" on page 282.
2. Enable DAI on a VLAN to inspect ARP packets.Refer to "Enabling DAI on a VLAN" on page 282.
3. Configure the trust settings of the VLAN members. ARP packets received on *trusted* ports bypass the DAI validation process. ARP packets received on *untrusted* ports go through the DAI validation process.Refer to "Enabling trust on a port" on page 283.
4. Enable DHCP snooping to populate the DHCP snooping IP-to-MAC binding database.

The following shows the default settings of DAI.

TABLE 70 Default DAI settings

Feature	Default
Dynamic ARP Inspection	Disabled
Trust setting for ports	Untrusted

Configuring an inspection ARP entry

Static ARP and static inspection ARP entries need to be configured for hosts on untrusted ports. Otherwise, when DAI checks ARP packets from these hosts against entries in the ARP table, it will not find any entries for them, and the Brocade device will not allow and learn ARP from an untrusted host.

When the inspection ARP entry is resolved with the correct IP/MAC mapping, its status changes from pending to valid.

To configure an inspection ARP entry, enter a command such as the following.

```
Brocade(config)# arp 10.20.20.12  0000.0002.0003 inspection
```

This command defines an inspection ARP entry, mapping a device IP address 10.20.20.12 with its MAC address 0000.0002.0003. The ARP entry will be in **Pend** (pending) status until traffic with the matching IP-to-MAC is received on a port.

Syntax: [**no**] **arp** *ip-addr mac-addr* **inspection**

The *ip-addr mac-addr* parameter specifies a device IP address and MAC address pairing.

Enabling DAI on a VLAN

DAI is disabled by default. To enable DAI on an existing VLAN, enter the following command.

```
Brocade(config)# ip arp inspection vlan 2
```

The command enables DAI on VLAN 2. ARP packets from untrusted ports in VLAN 2 will undergo DAI inspection.

Syntax: [**no**] **ip arp inspection vlan** *vlan-number*

The *vlan-number* variable specifies the ID of a configured VLAN.

Enabling trust on a port

The default trust setting for a port is untrusted. For ports that are connected to host ports, leave their trust settings as untrusted.

To enable trust on a port, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/1/4
Brocade(config-if-e10000-1/1/4)# arp inspection trust
```

The commands change the CLI to the interface configuration level of port 1/1/4 and set the trust setting of port 1/1/4 to trusted.

Syntax: [**no**] **arp inspection trust**

Displaying ARP inspection status and ports

To display the ARP inspection status for a VLAN and the trusted or untrusted port, enter the following command.

```
Brocade# show ip arp inspection vlan 2
IP ARP inspection VLAN 2: Disabled
  Trusted Ports :   ethe 1/1/4
  Untrusted Ports : ethe 1/1/1 to 1/1/3 ethe 1/2/1 to 1/2/4 ethe 1/3/1 to 1/3/4
ethe 1/2/7 to 1/2/9
```

Syntax: show ip arp inspection [**vlan** *vlan_id*]

The *vlan_id* variable specifies the ID of a configured VLAN.

Displaying the ARP table

To display the ARP table, enter the **show arp** command.

```
Brocade# show arp
Total number of ARP entries: 2, maximum capacity: 6000
No     IP Address          MAC Address      Type     Age      Port     Status
1      10.43.1.1           0000.00a0.4000   Dynamic  0        mgmt1    Valid
2      10.43.1.78          0000.0160.6ab1   Dynamic  2        mgmt1    Valid
```

The command displays all ARP entries in the system.

Syntax: show arp

DHCP snooping

Dynamic Host Configuration Protocol (DHCP) snooping enables the Brocade device to filter untrusted DHCP packets in a subnet. DHCP snooping can ward off MiM attacks, such as a malicious user posing as a DHCP server sending false DHCP server reply packets with the intention of misdirecting other users. DHCP snooping can also stop unauthorized DHCP servers and prevent errors due to user mis-configuration of DHCP servers.

Often DHCP snooping is used together with Dynamic ARP Inspection and IP Source Guard.

How DHCP snooping works

When enabled on a VLAN, DHCP snooping stands between untrusted ports (those connected to host ports) and trusted ports (those connected to DHCP servers). A VLAN with DHCP snooping enabled forwards DHCP request packets from clients and discards DHCP server reply packets on untrusted ports, and it forwards DHCP server reply packets on trusted ports to DHCP clients, as shown in the following figures

FIGURE 17 DHCP snooping at work - on an untrusted port

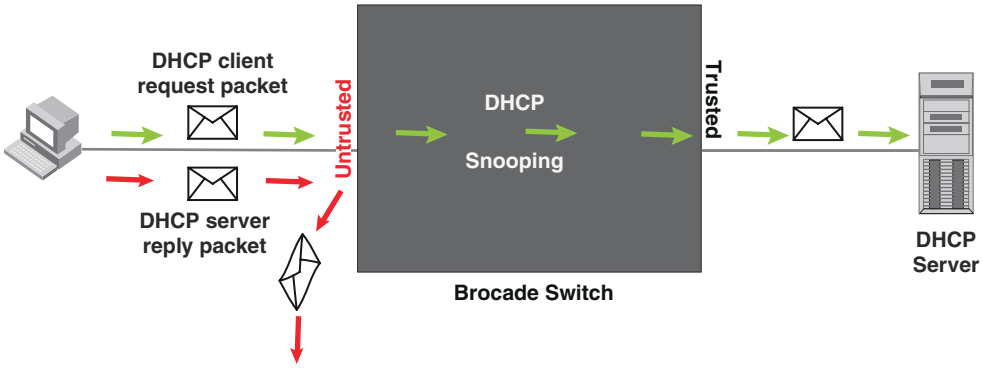
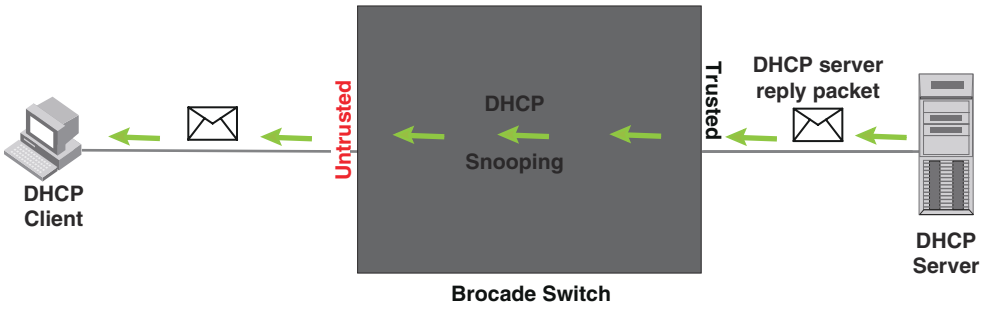


FIGURE 18 DHCP snooping at work - on a trusted port



DHCP binding database

When it forwards DHCP server reply packets on trusted ports, the Brocade device saves the client IP-to-MAC address binding information in the DHCP binding database. This is how the DHCP snooping binding table is populated. The information saved includes MAC address, IP address, lease time, VLAN number, and port number.

In the Brocade device, the DHCP binding database is integrated with the enhanced ARP table, which is used by Dynamic ARP Inspection. For more information, refer to "ARP entries" on page 280.

The lease time will be refreshed when the client renews its IP address with the DHCP server; otherwise the Brocade device removes the entry when the lease time expires.

Client IP-to-MAC address mappings

Client IP addresses need not be on directly-connected networks, as long as the client MAC address is learned on the client port and the client port is in the same VLAN as the DHCP server port. In this case, the system will learn the client IP-to-MAC port mapping. Therefore, a VLAN with DHCP snooping enabled does not require a VE interface.

In earlier releases, in the Layer 3 software image, DHCP snooping does not learn the secure IP-to-MAC address mapping for a client, if the client port is not a virtual ethernet (VE) interface with an IP subnet address. In other words, the client IP address had to match one of the subnets of the client port in order for DHCP to learn the address mapping.

System reboot and the binding database

To allow DAI and DHCP snooping to work smoothly across a system reboot, the binding database is saved to a file in the system flash memory after an update to the binding database, with a 30 second delay. The flash file is written and read only if DHCP snooping is enabled.

Configuration notes and feature limitations for DHCP snooping

The following limits and restrictions apply to DHCP snooping:

- To run DHCP snooping, you must first enable support for ACL filtering based on VLAN membership or VE port membership. To do so, enter the following commands at the Global CONFIG Level of the CLI.

```
Brocade(config)# enable ACL-per-port-per-vlan
Brocade(config)# write memory
Brocade(config)# exit
Brocade# reload
```

NOTE
You must save the configuration and reload the software to place the change into effect.

- DHCP snooping is supported on trunk ports (tagged and untagged) for trusted ports.
- DHCP snooping is not supported on trunk ports for untrusted ports.
- DHCP snooping is not supported together with DHCP Auto-configuration.
- A switch can have up to 256 ARP entries, therefore, DHCP entries are limited to 256. A router, however, can have 64,000 ARP entries, so a router can have up to 64,000 DHCP entries, of which only 1024 entries can be saved to flash on reboot.
- ACLs are supported on member ports of a VLAN on which DHCP snooping and Dynamic ARP Inspection (DAI) are enabled.
- See also "Client IP-to-MAC address mappings" on page 285.
- DHCP snooping supports DHCP relay agent information (DHCP Option 82). For details, refer to "DHCP relay agent information" on page 288.

Configuring DHCP snooping

Configuring DHCP snooping consists of the following steps.

1. Enable DHCP snooping on a VLAN.Refer to "Enabling DHCP snooping on a VLAN" on page 286.
2. For ports that are connected to a DHCP server, change their trust setting to trusted.Refer to "Enabling trust on a port" on page 286.

The following shows the default settings of DHCP snooping.

TABLE 71 Default DHCP snooping settings

Feature	Default
DHCP snooping	Disabled
Trust setting for ports	Untrusted

Enabling DHCP snooping on a VLAN

When DHCP snooping is enabled on a VLAN, DHCP packets are inspected.

DHCP snooping is disabled by default. This feature must be enabled on the client and the DHCP server VLANs. To enable DHCP snooping, enter the following global command for these VLANs.

```
Brocade(config)# ip dhcp snooping vlan 2
```

The command enables DHCP snooping on VLAN 2.

Syntax: [**no**] **ip dhcp snooping vlan** *vlan-number*

The *vlan-number* variable specifies the ID of a configured client or DHCP server VLAN.

Enabling trust on a port

The default trust setting for a port is untrusted. To enable trust on a port connected to a DHCP server, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/1/1
Brocade(config-if-e10000-1/1/1)# dhcp snooping trust
```

Port 1/1/1 is connected to a DHCP server. The commands change the CLI to the interface configuration level of port1/ 1/1 and set the trust setting of port 1/1/1 to trusted.

Syntax: [**no**] **dhcp snooping trust**

Disabling the learning of DHCP clients on a port

You can disable DHCP client learning on an individual port. To do so, enter commands such as the following.

```
Brocade(config)# interface ethernet 1/1/1
Brocade(config-if-e10000-1/1/1)# dhcp snooping client-learning disable
```

Syntax: [**no**] **dhcp snooping client-learning disable**

Use the **no** form of the command to re-enable DHCP client learning on a port once it has been disabled.

Clearing the DHCP binding database

You can clear the DHCP binding database using the CLI command **clear dhcp**. You can remove all entries in the database, or remove entries for a specific IP address only.

To remove all entries from the DHCP binding database, enter the **clear dhcp** command.

```
Brocade# clear dhcp
```

To clear entries for a specific IP address, enter a command such as the following.

```
Brocade# clear dhcp 10.10.102.4
```

Syntax: clear dhcp [*ip-addr*]

Displaying DHCP snooping status and ports

To display the DHCP snooping status for a VLAN and the trusted/untrusted port, use the **show ip dhcp snooping vlan** command.

```
Brocade# show ip dhcp snooping vlan 2
IP DHCP snooping VLAN 2: Enabled
```

Syntax: show ip dhcp snooping [**vlan** *vlan-id*]

Displaying the DHCP snooping binding database

To display the DHCP snooping binding database, use the **show ip dhcp snooping** info command.

```
Brocade# show ip dhcp snooping info
Dhcp snooping Info
Total learnt entries 1
SAVED DHCP ENTRIES IN FLASH
       IP Address       Mac Address     Port  vlan  lease
0      10.10.10.20      0000.0002.0003  1/1/3  1112  361
```

Syntax: show ip dhcp snooping info

Displaying DHCP binding entry and status

To display the DHCP binding entry and its current status, use the **show arp** command.

```
Brocade# show arp
Total number of ARP entries: 2, maximum capacity: 6000
No.   IP Address          MAC Address      Type     Age      Port     Status
1     10.43.1.1           0000.0001.c320   Dynamic  0        mgmt1    Valid
2     10.43.1.199         0000.0002.b263   Dynamic  7        mgmt1    Valid
```

Syntax: show arp

DHCP snooping configuration example

The following example configures VLAN 2 and VLAN 20, and changes the CLI to the global configuration level to enable DHCP snooping on the two VLANs. The commands are as follows.

```
Brocade(config)# vlan 2
Brocade(config-vlan-2)# untagged ethe 1/1/3 to 1/1/4
Brocade(config-vlan-2)# router-interface ve 2
Brocade(config-vlan-2)# exit
Brocade(config)# ip dhcp snooping vlan 2

Brocade(config)# vlan 20
Brocade(config-vlan-20)# untagged ethe 1/1/1 to 1/1/2
Brocade(config-vlan-20)# router-interface ve 20
Brocade(config-vlan-20)# exit
Brocade(config)# ip dhcp snooping vlan 20
```

On VLAN 2, client ports 1/1/3 and 1/1/4 are untrusted by default all client ports are untrusted. Hence, only DHCP client request packets received on ports 1/1/3 and 1/1/4 are forwarded.

On VLAN 20, ports 1/1/1 and 1/1/2 are connected to a DHCP server. DHCP server ports are set to trusted.

```
Brocade(config)# interface ethernet 1/1/1
Brocade(config-if-e10000-1/1/1)# dhcp snooping trust
Brocade(config-if-e10000-1/1/1)# exit
Brocade(config)# interface ethernet 1/1/2
Brocade(config-if-e10000-1/1/2)# dhcp snooping trust
Brocade(config-if-e10000-1/1/2)# exit
```

Hence, DHCP server reply packets received on ports 1/1/1 and 1/1/2 are forwarded, and client IP/MAC binding information is collected.

The example also sets the DHCP server address for the local relay agent.

```
Brocade(config)# interface ve 2
Brocade(config-vif-2)# ip address 10.20.20.1/24
Brocade(config-vif-2)# ip helper-address 1 10.30.30.4
Brocade(config-vif-2)# interface ve 20
Brocade(config-vif-20)# ip address 10.30.30.1/24
```

DHCP relay agent information

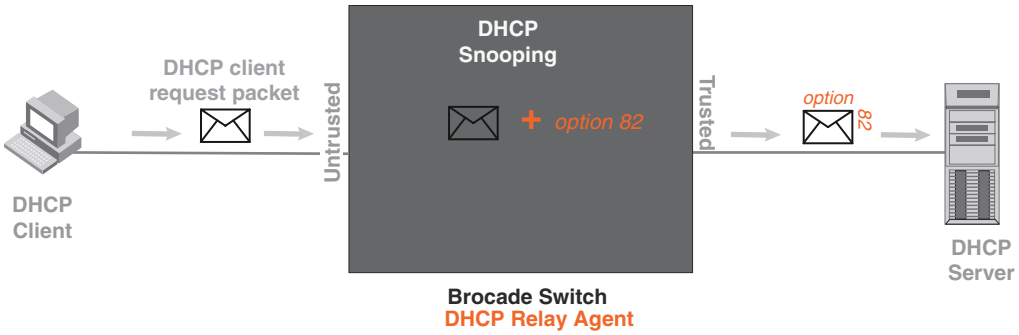
DHCP relay agent information, also known as DHCP option 82, enables a DHCP relay agent to insert information about a clients' identity into a DHCP client request being sent to a DHCP server.

When DHCP snooping is enabled, DHCP option 82 is automatically enabled. DHCP packets are processed as follows:

- Before relaying a DHCP discovery packet or DHCP request packet from a client to a DHCP server, the Brocade ICX 6650 will add agent information to the packet.
- Before relaying a DHCP reply packet from a DHCP server to a client, the Brocade ICX 6650 will remove relay agent information from the packet.

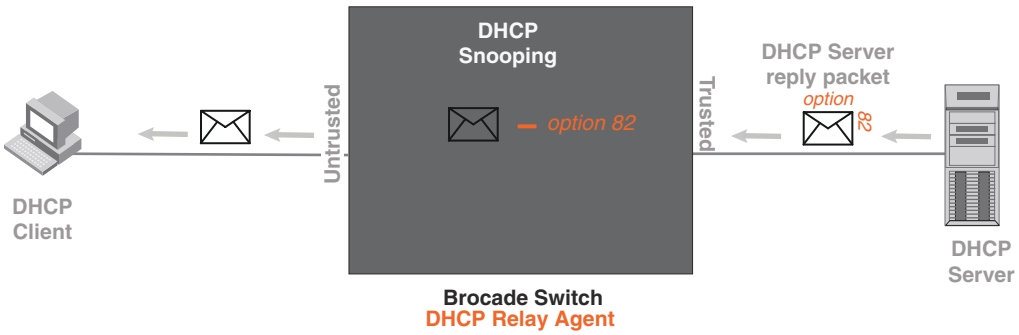
As illustrated in Figure 19, the DHCP relay agent (the Brocade ICX 6650 switch), inserts DHCP option 82 attributes when relaying a DHCP request packet to a DHCP server.

FIGURE 19 DHCP Option 82 attributes added to the DHCP packet



As illustrated in Figure 20, the Brocade ICX 6650 device deletes DHCP option 82 attributes before forwarding a server reply packet back to a DHCP client.

FIGURE 20 DHCP Option 82 attributes removed from the DHCP packet



The DHCP option 82 insertion/deletion feature is available only when DHCP snooping is enabled for the client/server ports.

Configuration notes for DHCP option 82

- DHCP snooping and DHCP option 82 are supported on a per-VLAN basis.
- DHCP option 82 follows the same configuration rules and limitations as for DHCP snooping. For more information, refer to "Configuration notes and feature limitations for DHCP snooping" on page 285.

DHCP option 82 sub-options

The Brocade implementation of DHCP Option 82 supports the following sub-options:

- Sub-Option 1 – Circuit ID
- Sub-Option 2 - Remote ID
- Sub-Option 6 – Subscriber ID

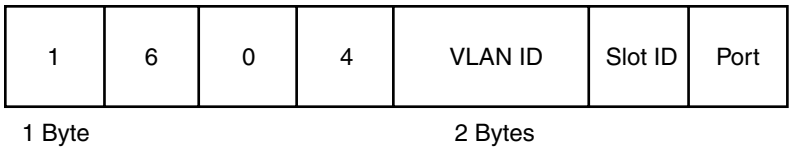
These sub-options are described in the following sections.

Sub-option 1 – Circuit ID

The Circuit ID (CID) identifies the circuit or port from which a DHCP client request was sent. The Brocade ICX 6650 device uses this information to relay DHCP responses back to the proper circuit, for example, the port number on which the DHCP client request packet was received.

Brocade ICX 6650 devices support the ***General CID packet format***. This simple format encodes the CID type, actual information length, VLAN ID, slot number, and port number. This format is compatible with the format used by other vendors' devices. Figure 21 illustrates the general CID packet format.

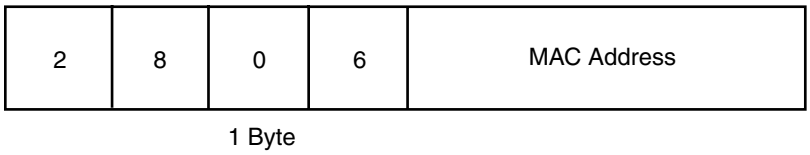
FIGURE 21 General CID packet format



Sub-option 2 – Remote ID

The Remote ID (RID) identifies the remote host end of the circuit (the relay agent). Brocade devices use the MAC address to identify itself as the relay agent. Figure 22 illustrates the RID packet format.

FIGURE 22 RID packet format



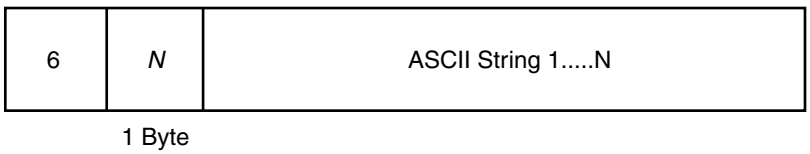
Sub-option 6 - Subscriber ID

The Subscriber ID (SID) is a unique identification number that enables an Internet Service Provider to:

- Identify a subscriber
- Assign specific attributes to that subscriber (for example, host IP address, subnet mask, and domain name server (DNS))
- Trigger accounting

Figure 23 illustrates the SID packet format.

FIGURE 23 SID packet format



The second byte (*N* in Figure 23) is the length of the ASCII string that follows. The Brocade ICX 6650 device supports up to 50 ASCII characters.

DHCP option 82 configuration

When DHCP snooping is enabled on a VLAN, DHCP option 82 also is enabled by default. You do not need to perform any extra configuration steps to enable this feature. To enable DHCP snooping, refer to"Enabling DHCP snooping on a VLAN" on page 286.

When processing DHCP packets, the Brocade ICX 6650 device applies the following default behavior when DHCP option 82 is enabled:

- Subjects all ports in the VLAN to DHCP option 82 processing
- Uses the general CID packet format
- Uses the standard RID packet format
- Replaces relay agent information received in DHCP packets with its own information
- Does not enable SID processing

When DHCP option 82 is enabled, you can optionally:

- Disable DHCP Option 82 processing on individual ports in the VLAN
- Configure the device to drop or keep the relay agent information in a DHCP packet instead of replacing it with its own information
- Enable SID processing

Disabling and re-enabling DHCP option 82 processing on an individual interface

By default, when DHCP option 82 is enabled on a VLAN, DHCP packets received on all member ports of the VLAN are subject to DHCP option 82 processing. You can optionally disable and later re-enable DHCP option 82 processing on one or more member ports of the VLAN. To do so, use the commands in this section.

To disable a particular port in a VLAN from adding relay agent information to DHCP packets, enter commands such as the following.

```
Brocade(config)# ip dhcp snooping vlan 1
Brocade(config)# interface ethernet 1/1/4
Brocade(config-if-e1000-1/1/4)# no dhcp snooping relay information
```

The first CLI command enables DHCP snooping and DHCP option 82 on VLAN 1. The second command changes the CLI configuration level to the Interface configuration level for port e 1/1/4. The last command disables DHCP option 82 on interface e 1/1/4, which is a member of VLAN 1.

To re-enable DHCP option 82 on an interface after it has been disabled, enter the following command at the Interface level of the CLI.

```
Brocade(config-if-e10000-1/1/4)# dhcp snooping relay information
```

Syntax: [no] **dhcp snooping relay information**

Use the **show ip dhcp snooping vlan** command to view the ports on which DHCP option 82 processing is disabled. For more information, refer to "Viewing the ports on which DHCP option 82 is disabled" on page 293.

Changing the forwarding policy

When the Brocade device receives a DHCP message that contains relay agent information, by default, the device replaces the information with its own relay agent information. If desired, you can configure the device to keep the information instead of replacing it, or to drop (discard) messages that contain relay agent information. To do so, use the CLI commands in this section.

For example, to configure the device to *keep* the relay agent information contained in a DHCP message, enter the **ip dhcp relay information policy keep** command.

```
Brocade(config)# ip dhcp relay information policy keep
```

To configure the device to *drop* DHCP messages that contain relay agent information, enter the **ip dhcp relay information policy drop** command.

```
Brocade(config)# ip dhcp relay information policy drop
```

Syntax: ip dhcp relay information policy *policy-type*

policy-type can be one of the following:

- **drop** – Configures the device to discard messages containing relay agent information
- **keep** – Configures the device to keep the existing relay agent information
- **replace** – Configures the device to overwrite the relay agent information with the information in the Brocade configuration. This is the default behavior.

Use the **show ip dhcp relay information** command to view the forwarding policy configured on the switch.Refer to "Viewing the circuit ID, remote ID, and forwarding policy" on page 293.

Enabling and disabling subscriber ID processing

You can configure a unique subscriber ID (SID) per port. Unlike the CID and RID sub-options, the SID sub-option is not automatically enabled when DHCP option 82 is enabled. To enable SID processing, enter commands such as the following.

```
Brocade(config)# ip dhcp snooping vlan 1
Brocade(config)# interface ethernet 1/1/4
Brocade(config-if-e10000-1/1/4)# dhcp snooping relay information subscriber-id
Brcd01
```

The first CLI command enables DHCP snooping and DHCP option 82 on VLAN 1. The second command changes the CLI configuration level to the Interface configuration level for port e 1/1/4. The last command enables interface e 1/1/4 to insert the SID information in DHCP packets. In this case, the SID is **Brcd01**. All other ports in VLAN 1 on which SID is not enabled will send the standard relay agent information (CID and RID information) only.

Syntax: [**no**] **dhcp snooping relay information option subscriber-id** *ASCII-string*

Enter up to 50 alphanumeric characters for *ASCII-string*

Use the **no** form of the command to disable SID processing once it is enabled.

Use the **show interfaces ethernet** command to view the subscriber ID configured on a port.Refer to "Viewing the status of DHCP option 82 and the subscriber ID" on page 294.

Viewing information about DHCP option 82 processing

Use the commands in this section to view information about DHCP option 82 processing.

Viewing the circuit ID, remote ID, and forwarding policy

Use the **show ip dhcp relay information** command to obtain information about the circuit ID, remote ID, and forwarding policy for DHCP option 82. The following shows an example output.

```
Brocade# show ip dhcp relay information
Relay Information: Format: Circuit-ID : vlan-mod-port
                           Remote-ID : mac
                              Policy : keep
```

Syntax: show ip dhcp relay information

TABLE 72 Output for the ip dhcp relay information command

Field	Description
Circuit-ID	The agent circuit ID format: • **vlan-mod-port** – The default circuit ID format.
Remote-ID	The remote ID format. This field displays **mac**, which is the default remote ID format.
Policy	How the Brocade switch processes relay agent information it receives in DHCP messages: • **drop** – drops the relay agent information • **keep** – keeps the relay agent information • **replace** – replaces the relay agent information with its own

Viewing the ports on which DHCP option 82 is disabled

Use the following command to refer which port in a DHCP snooping VLAN has DHCP Option 82 disabled.

```
Brocade# show ip dhcp snooping vlan 1
IP DHCP snooping VLAN 1: Enabled
  Trusted Ports :   ethe 1/1/3
  Untrusted Ports : ethe 1/1/1 to 1/1/2 ethe 1/1/4 to 1/1/24
  Relay Info. disabled Ports: ethe 1/1/10
```

Syntax: show ip dhcp snooping vlan *vlan-id*

TABLE 73 Output for the **show ip dhcp snooping vlan** command

Field	Description
IP DHCP snooping VLAN *vlan-id*	The DHCP snooping and DHCP option 82 status for a VLAN: • Enabled • Disabled
Trusted Ports	A list of trusted ports in the VLAN.

TABLE 73 Output for the **show ip dhcp snooping vlan** command

Field	Description
Untrusted Ports	A list of untrusted ports in the VLAN.
Relay Info. disabled Ports	Ports on which DHCP option 82 was disabled.

Viewing the status of DHCP option 82 and the subscriber ID

Use the **show interfaces ethernet** command to obtain information about the status of DHCP option 82 and the configured subscriber ID, if applicable. In the example below, the text in **bold** type displays the information specific to DHCP option 82.

```
Brocade# show interfaces ethernet 1/1/3
Ethernet3 is up, line protocol is up
  Hardware is Ethernet, address is 0000.0020.0002 (bia 00e0.5200.0002)
  Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
  Configured mdi mode AUTO, actual MDI
  Member of L2 VLAN ID 1, port is untagged, port state is FORWARDING
  BPDU guard is Disabled, ROOT protect is Disabled
  Link Error Dampening is Disabled
  STP configured to ON, priority is level0
  Flow Control is config enabled, oper enabled, negotiation disabled
  mirror disabled, monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  IPG MII 96 bits-time, IPG GMII 96 bits-time
  IP MTU 1500 bytes
  300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 264 bits/sec, 0 packets/sec, 0.00% utilization
  0 packets input, 0 bytes, 0 no buffer
  Received 0 broadcasts, 0 multicasts, 0 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runts, 0 giants
  0 packets output, 0 bytes, 0 underruns
  Transmitted 0 broadcasts, 0 multicasts, 0 unicasts
  0 output errors, 0 collisions
     Relay Agent Information option: Enabled, Subscriber-ID: Brocade001
```

The above output shows that DHCP option 82 is **Enabled** on the device and the configured subscriber ID is **Brocade001**.

Syntax: show interfaces ethernet *port*

Specify the port variable in the *stack-unit/slotnum/portnum* format.

IP source guard

You can use IP Source Guard together with Dynamic ARP Inspection on untrusted ports. Refer to "DHCP snooping" on page 283 and "Dynamic ARP inspection" on page 279.

The Brocade implementation of the IP Source Guard feature supports configuration on a port, on specific VLAN memberships on a port (Layer 2 devices only), and on specific ports on a virtual interface (VE) (Layer 3 devices only).

When IP Source Guard is first enabled, only DHCP packets are allowed and all other IP traffic is blocked. When the system learns a valid IP address, IP Source Guard then allows IP traffic. Only the traffic with valid source IP addresses are permitted. The system learns of a valid IP address from DHCP Snooping. When it learns a valid IP address, the system permits the learned source IP address.

When a new IP source entry binding on the port is created or deleted, the ACL will be recalculated and reapplied in hardware to reflect the change in IP source binding. By default, if IP Source Guard is enabled without any IP source binding on the port, an ACL that denies all IP traffic is loaded on the port.

Configuration notes and feature limitations for IP source guard

- To run IP Source Guard, you must first enable support for ACL filtering based on VLAN membership or VE port membership. To do so, enter the following commands at the Global CONFIG Level of the CLI.

```
Brocade(config)# enable ACL-per-port-per-vlan
Brocade(config)# write memory
Brocade(config)# exit
Brocade# reload
```

NOTE
You must save the configuration and reload the software to place the change into effect.

- Brocade devices support IP Source Guard together with IPv4 ACLs (similar to ACLs for Dot1x), as long as both features are configured at the port-level or per-port-per-VLAN level. Brocade devices do not support IP Source Guard and IPv4 ACLs on the same port if one is configured at the port-level and the other is configured at the per-port-per-VLAN level.
- IP source guard and IPv6 ACLs are supported together on the same device, as long as they are not configured on the same port or virtual Interface.
- The following limitations apply when configuring IP Source Guard on Layer 3 devices:
  - You cannot enable IP Source Guard on a tagged port on a Layer 3 device. To enable IP Source Guard on a tagged port, enable it on a per-VE basis.
  - You cannot enable IP Source Guard on an untagged port with VE on a Layer 3 device. To enable IP Source Guard in this configuration, enable it on a per-VE basis.
  - There are no restrictions for Layer 2, either on the port or per-VLAN.
- You cannot enable IP Source Guard on a port that has any of the following features enabled:
  - MAC address filter
  - Rate limiting
  - Trunk port
  - 802.1x with ACLs
  - Multi-device port authentication with ACLs
- A port on which IP Source Guard is enabled limits the support of IP addresses, VLANs, and ACL rules per port. An IP Source Guard port supports a maximum of:
  - 64 IP addresses
  - 64 VLANs

- 64 rules per ACL

- The number of configured ACL rules affect the rate at which hardware resources are used when IP Source Guard is enabled. Use the **show access-list hw-usage on** command to enable hardware usage for an ACL, followed by a show access-list *access-list-id* command to determine the hardware usage for an ACL.

  Example

  ```
  Brocade# show access-list hw-usage on
  Brocade# show access-list 100
  Extended IP access list 100 (hw usage : 2)
  deny ip any any (hw usage : 1)
  ```

  To provide more hardware resource for IP Source Guard addresses, modify the ACL rules so that it uses less hardware resource.

- If you enable IP Source Guard in a network topology that has DHCP clients, you must also enable DHCP snooping. Otherwise, all IP traffic including DHCP packets will be blocked.
- When you enable IP Source Guard in a network topology that does not have DHCP clients, you must create an IP source binding for each client that will be allowed access to the network. Otherwise, data packets will be blocked. Refer to "Defining static IP source bindings" on page 296.
- Source Guard Protection enables concurrent support with multi-device port authentication. For details, Refer to "Enabling source guard protection" on page 246.
- IP Source Guard is supported on a VE with or without an assigned IP address.

Enabling IP source guard on a port

You can enable IP Source Guard on DHCP snooping untrusted ports. Refer to "DHCP snooping" on page 283 for how to configure DHCP and DHCP untrusted ports.

By default, IP Source Guard is disabled. To enable IP Source Guard on a DHCP untrusted port, enter the following commands.

```
Brocade(config)# interface ethernet 1/1/4
Brocade(config-if-e10000-1/1/4)# source-guard enable
```

The commands change the CLI to the interface configuration level for port 1/1/4 and enable IP Source Guard on the port.

Syntax: [**no**] **source-guard enable**

Defining static IP source bindings

You can manually enter valid IP addresses in the binding database. To do so, enter a command such as the following.

```
Brocade(config)# ip source binding 10.10.10.1 ethernet 1/2/4 vlan 4
```

Syntax: [**no**] **ip source binding** *ip-addr* **ethernet** *stack-iunit/slotnum/portnum* [**vlan** *vlannum*]

For *ip-addr*, enter a valid IP address.

The [vlan *vlannum*] parameter is optional. If you enter a VLAN number, the binding applies to that VLAN only. If you do not enter a VLAN number, the static binding applies to all VLANs associated with the port. Note that since static IP source bindings consume system resources, you should avoid unnecessary bindings.

Enabling IP source guard per-port-per-VLAN

To enable IP Source Guard per-port-per-VLAN, enter commands such as the following.

```
Brocade(config)# vlan 12 name vlan12
Brocade(config-vlan-12)# untag ethernet 1/1/5 to 1/1/8
Brocade(config-vlan-12)# tag ethernet 1/1/23 to 1/1/24
Brocade(config-vlan-12)# exit
Brocade(config)# interface ethernet 1/1/23
Brocade(config-if-e10000-1/1/23)# per-vlan vlan12
Brocade(config-if-e10000-1/1/23-vlan-12))# source-guard enable
```

The commands in this example configure port-based VLAN 12, and add ports e 1/1/5 – 1/1/8 as untagged ports and ports e 1/1/23 –1/1/24 as tagged ports to the VLAN. The last two commands enable IP Source Guard on port e 1/1/23, a member of VLAN 12.

Syntax: [no] **source-guard enable**

Enabling IP source guard on a VE

To enable IP Source Guard on a virtual interface, enter commands such as the following.

```
Brocade(config)# vlan 2
Brocade(config-vlan-2)# tag e1/1/1
Added tagged port(s) ethe 1/1/1 to port-vlan 2
Brocade(config-vlan-2)# router-int ve 2
Brocade(config-vlan-2)# int ve 2
Brocade(config-vif-2)# source-guard enable ethernet 1/1/1
```

Syntax: [no] **source-guard enable**

Displaying learned IP addresses

To display the learned IP addresses for IP Source Guard ports, use the CLI commands **show ip source-guard ethernet**.

Chapter

13

Limiting Broadcast, Multicast, and Unknown Unicast Traffic

This chapter describes how rate limiting for broadcast, multicast, and unknown-unicast traffic is implemented and configured on Brocade ICX 6650.

Broadcast, unknown Unicast, and Multicast rate limiting

Brocade devices forward all flooded traffic at wire speed within a VLAN. However, some third-party networking devices cannot handle high rates of broadcast, multicast, or unknown-unicast traffic. If not controlled such scenarios can result in Denial of Service (DoS).

To control the such traffic from being forwarded to other devices in a VLAN, you can limit the number of broadcast, multicast, or unknown-unicast (BUM) packets received each second on every port of Brocade ICX 6650.

Configuration notes and feature limitations

- Brocade ICX 6650 supports packet-based limiting only. Limits set on such flooded traffic are also in terms of packets per second.

Configuring rate limiting for BUM traffic

To enable broadcast limiting on a group of ports by counting the number of packets received, enter the following commands:

```
Brocade(config)# interface ethernet 1/1/1 to 1/1/8
Brocade(config-mif-1/1/1-1/1/8)# broadcast limit 65536
```

To include unknown-unicast limiting, enter the **unknown-unicast limit** command after enabling broadcast limiting.

```
Brocade(config-mif-1/1/1-1/1/8)# unknown-unicast limit
```

To include multicast limiting, enter the **multicast limit** command after enabling broadcast limiting.

```
Brocade(config-mif-1/1/1-1/1/8)# multicast limit
```

Syntax: [no]broadcast limit *num*

Syntax: [no] multicast limit

Syntax: [no] unknown-unicast limit

The *num* variable specifies the maximum number of packets per second. It can be any number that is a multiple of 65536, up to a maximum value of 2147418112 for 10G and multiple of 8192 for 1G. If you enter the **multicast limit** or **unknown-unicast limit** command, multicast or unknown-unicast packets are included in the corresponding limit. If you specify 0, limiting is disabled. If you specify a number that is not a multiple of 65536, the software rounds the number to the next multiple of 65536. Limiting is disabled by default.

Viewing rate limits set on BUM traffic

You can use the **show run interface** command to display the broadcast, multicast, and unknown-unicast limits configured on each interface of the device. In addition to **show run interface**, you can use the **show rate-limit broadcast** command to display the broadcast, multicast, and unknown-unicast limits configured on the device.

Use the **show run interface** command to view the broadcast, multicast, and unknown-unicast limit configured on each interface as shown in the following example.

Example

```
Brocade# show run interface
interface management 1
 ip address 10.21.113.7 255.255.248.0
!
interface ethernet 1/1/1
 broadcast limit 65536
 multicast limit
 unknown-unicast limit
!
interface ethernet 1/1/2
 broadcast limit 65536
 multicast limit
 unknown-unicast limit
!
interface ethernet 1/1/3
 broadcast limit 65536
 multicast limit
 unknown-unicast limit
!
interface ethernet 1/1/4
 broadcast limit 65536
 multicast limit
 unknown-unicast limit
!
interface ethernet 1/1/5
 broadcast limit 65536
 multicast limit
 unknown-unicast limit
!
interface ethernet 1/1/6
 broadcast limit 65536
 multicast limit
 unknown-unicast limit
!
interface ethernet 1/1/7
 broadcast limit 65536
 multicast limit
 unknown-unicast limit
!
```

```
interface ethernet 1/1/8
 broadcast limit 65536
 multicast limit
 unknown-unicast limit
!
interface ethernet 1/2/1
 optical-monitor 0
!
!
interface tunnel 1
!
```

Syntax: show run interface

Use the **show rate-limit broadcast** command to display the broadcast, multicast, and unknown-unicast limit configured for each port region to which it applies.

Example

```
Brocade# show rate-limit broadcast
Broadcast/Multicast/Unknown Unicast Limit Settings:
Port            Limit  Packets/Bytes   Packet Type(s)
1/1/1           65536        Packets   Broadcast + Multicast + Unknown Unicast
1/1/2           65536        Packets   Broadcast + Multicast + Unknown Unicast
1/1/3           65536        Packets   Broadcast + Multicast + Unknown Unicast
1/1/4           65536        Packets   Broadcast + Multicast + Unknown Unicast
1/1/5           65536        Packets   Broadcast + Multicast + Unknown Unicast
1/1/6           65536        Packets   Broadcast + Multicast + Unknown Unicast
1/1/7           65536        Packets   Broadcast + Multicast + Unknown Unicast
1/1/8           65536        Packets   Broadcast + Multicast + Unknown Unicast
```

Syntax: show rate-limit broadcast

Broadcast, unknown Unicast, and Multicast rate limiting

Index

Numerics

A

B

C

D

F

G

I

L

M

P

Q

R

S

T

U

V