BROCADE

CONFIGURATION GUIDE

Brocade 5600 vRouter Remote Management Configuration Guide

Supporting Brocade 5600 vRouter 4.2R1

53-1004273-01
16 May 2016

Contents

Preface

Document conventions

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

Text formatting conventions

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

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

Command syntax conventions

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

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

Convention	Description
...	Repeat the previous element, for example, *member*[*member*...].
\	Indicates a "soft" line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Notes, cautions, and warnings

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

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

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

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

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

Brocade resources

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

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

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

Release notes are available on MyBrocade under Product Downloads.

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

Contacting Brocade Technical Support

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

Brocade customers

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

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

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

Brocade OEM customers

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

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

Document feedback

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

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

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

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

About This Guide

This guide describes how to configure SSH, Telnet, Web GUI access (HTTPS), NETCONF and SNMP for remote management of the Brocade 5600 vRouter (referred to as a virtual router, vRouter, or router in the guide).

For information about how to remotely manage the vRouter by using the Brocade 5600 vRouter Remote Access API, a REST API, refer to *Brocade 5600 vRouter Remote Access API Reference Guide*.

SSH

SSH configuration

Secure Shell (SSH) provides a secure mechanism to log on to the Brocade vRouter and access the Command Line Interface (CLI). Configuring SSH is optional, but is recommended to provide secure remote access to the Brocade vRouter. In addition to the standard password authentication provided by SSH, shared public key authentication is also available.

The following table enables SSH for password authentication on the default port (port 22), as shown in the following figure. By default, only SSH version 2 is enabled.

FIGURE 1 Enabling SSH access

To enable the SSH service on the Brocade vRouter, perform the following steps in configuration mode.

TABLE 1 Enabling SSH access

Step	Command
Create the configuration node for the SSH service.	`vyatta@R1# set service ssh`
Commit the information	`vyatta@R1# commit` `Restarting OpenBSD Secure Shell server: sshd.`

Show the configuration.	`vyatta@R1# show service`
	` ssh {`
	` }`

SSH Commands

service ssh allow-root

Specifies that root logins are to be allowed on SSH connections.

Syntax

set service ssh allow-root

delete service ssh allow-root

show service ssh

Command Default

Root logins are not allowed on SSH connections.

Modes

Configuration mode

Configuration Statement

```
service {
    ssh {
        allow-root
    }
}
```

Usage Guidelines

Use this command to specify that root logins are to be allowed on SSH connections.

> NOTE
> The **root** account is often the target of external attacks so its use is discouraged. The **vyatta** account provides sufficient privileges to administer the system.

Use the **set** form of this command to specify that root logins are to be allowed on SSH connections.

Use the **delete** form of this command to restore the default allow-root configuration.

Use the **show** form of this command to view the configuration.

service ssh

Enables SSH as an access protocol on the Brocade vRouter.

Syntax

set service ssh

delete service ssh

show service ssh

Modes

Configuration mode

Configuration Statement

```
service {
    ssh {
    }
}
```

Usage Guidelines

Use this command to configure the system to allow SSH requests from remote systems to the local system.

Creating the SSH configuration node enables SSH as an access protocol. By default, the router uses port 22 for the SSH service, and SSH version 2 alone is used.

Use the **set** form of this command to create the SSH configuration.

Use the **delete** form of this command to remove the SSH configuration. If you delete the SSH configuration node you will disable SSH access to the system.

Use the **show** form of this command to view the SSH configuration.

service ssh disable-host-validation

Specifies that SSH should not validate clients via reverse DNS lookup.

Syntax

set service ssh disable-host-validation

delete service ssh disable-host-validation

show service ssh

Command Default

Client PTR/reverse-DNS records are resolved through DNS.

Modes

Configuration mode

Configuration Statement

```
service {
    ssh {
        disable-host-validation
    }
}
```

Usage Guidelines

Use this command to specify that SSH should not resolve client PTR/reverse-DNS records via a reverse DNS (PTR) lookup. This process can be time consuming and cause long delays for clients trying to connect.

Use the **set** form of this command to specify that SSH should not resolve client PTR/reverse-DNS records via a reverse DNS (PTR) lookup.

Use the **delete** form of this command to restore the default configuration and allow reverse DNS lookups.

Use the **show** form of this command to view the configuration.

service ssh disable-password-authentication

Specifies that SSH users are not to be authenticated using passwords.

Syntax

set service ssh disable-password-authentication

delete service ssh disable-password-authentication

show service ssh

Command Default

Users are authenticated using passwords.

Modes

Configuration mode

Configuration Statement

```
service {
    ssh {
        disable-password-authentication
    }
}
```

Usage Guidelines

Use this command to specify that SSH users are not to be authenticated using passwords. This is typically done in order for SSH users to be authenticated using shared public keys instead. Shared public key authentication is less susceptible to brute force guessing of common passwords. If password authentication is disabled then shared public keys must be configured for user authentication. For information on configuring public keys for user authentication see *Brocade 5600 vRouter Basic System Configuration Guide*.

Use the **set** form of this command to specify that users are not to be authenticated by using passwords.

Use the **delete** form of this command to restore the default configuration and allow authentication by passwords.

Use the **show** form of this command to view the configuration.

service ssh listen-address <ipv4>

Configures access to SSH on a specific address.

Syntax

set service ssh listen-address *ipv4*

delete service ssh listen-address *ipv4*

show service ssh listen-address

Command Default

Requests to access SSH will be accepted on any system IP address.

Parameters

ipv4

Multi-node. An IP address that the ssh service listens for connection requests on. The address must be assigned to an interface.
You can define more than one **listen-address** by creating multiple **listen-address** configuration nodes.

Modes

Configuration mode

Configuration Statement

```
service {
    ssh {
        listen-address ipv4
    }
}
```

Usage Guidelines

Use this command to configure the system to accept requests for SSH access on specific addresses. This provides a way to limit access to the system.

Use the **set** form of this command to configure the system to accept requests for SSH access on specific addresses.

Use the **delete** form of this command to remove a listen-address.

Use the **show** form of this command to view the listen-address configuration.

service ssh port <port>

Specifies the port the system will use for the SSH service.

Syntax

set service ssh port *port*

delete service ssh port

show service ssh port

Command Default

The SSH service runs on port 22.

Parameters

port

The port the system uses for the SSH service. The numbers range from 1 through 65534.

Modes

Configuration mode

Configuration Statement

```
service {
    ssh {
        port port
    }
}
```

Usage Guidelines

Use this command to specify the port the system will use for the SSH service.

Use the **set** form of this command to specify the port the system will use for the SSH service.

Use the **delete** form of this command to restore the default port configuration.

Use the **show** form of this command to view the port configuration.

service ssh protocol-version <version>

Specifies which versions of SSH are enabled.

Syntax

set service ssh protocol-version *version*

delete service ssh protocol-version

show service ssh protocol-version

Command Default

SSH version 2 alone is enabled.

Parameters

version

Specifies which versions of SSH are enabled. Supported values are as follows:

v1

SSH version 1 alone is enabled.

v1

SSH version 1 alone is enabled.

v2

SSH version 2 alone is enabled. This is the recommended setting as **v1** is considered insecure.

all

Both SSH version 1 and SSH version 2 are both enabled.

Modes

Configuration mode

Configuration Statement

```
service {
    ssh {
        protocol-version version
    }
}
```

Usage Guidelines

Use this command to specify which versions of SSH are enabled.

Use the **set** form of this command to specify which versions of SSH are enabled.

Use the **delete** form of this command to restore the default protocol-version configuration.

Use the **show** form of this command to view the protocol-version configuration.

Telnet

Telnet configuration

Configuring Telnet is optional, but creating the Telnet service will allow you to access the Brocade vRouter remotely. The following table enables Telnet on the default port (port 23), as shown in the following figure.

FIGURE 2 Enabling telnet access

To enable the Telnet service on the Brocade vRouter, perform the following steps in configuration mode.

TABLE 2 Enabling telnet access

Step	Command
Create the configuration node for the Telnet service.	`vyatta@R1# set service telnet`
Commit the information.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show service`
	`  telnet {`
	`  }`

Telnet Commands

service telnet

Configures Telnet as an access protocol on the system.

Syntax

set service telnet

delete service telnet

show service telnet

Modes

Configuration mode

Configuration Statement

```
service {
    telnet {
    }
}
```

Usage Guidelines

Use this command to configure the system to accept Telnet as an access service to the system.

Creating the Telnet configuration node enables Telnet as an access protocol. By default, the system uses port 23 for the Telnet service.

Use the **set** form of this command to create the Telnet configuration.

Use the **delete** form of this command to remove the Telnet configuration. If you delete the Telnet configuration node you will disable Telnet access to the system.

Use the **show** form of this command to view the Telnet configuration.

service telnet allow-root

Specifies that root logins are allowed on Telnet connections.

Syntax

set service telnet allow-root

delete service telnet allow-root

show service telnet

Command Default

Root logins are not allowed on Telnet connections.

Modes

Configuration mode

Configuration Statement

```
service {
    telnet {
        allow-root
    }
}
```

Usage Guidelines

Use this command to specify that root logins are to be allowed on Telnet connections.

Use the **set** form of this command to specify that root logins are to be allowed on Telnet connections.

Use the **delete** form of this command to restore the default allow-root configuration.

Use the **show** form of this command to view the configuration.

service telnet listen-address <address>

Configures access to Telnet on a specific address.

Syntax

set service telnet listen-address *address*

delete service telnet listen-address *address*

show service telnet

Command Default

Requests to access Telnet will be accepted on any system IP address.

Parameters

listen-address *address*

Multi-node. An IPv4 or IPv6 address that the telnet service listens for connection requests on. The address must be assigned to an interface.

You can define more than one **listen-address** by creating multiple **listen-address** configuration nodes.

Modes

Configuration mode

Configuration Statement

```
service {
    telnet {
        listen-address address
    }
}
```

Usage Guidelines

Use this command to configure the system to accept requests for Telnet access on specific addresses. This provides a way to limit access to the system.

Use the **set** form of this command to configure the system to accept requests for Telnet access on specific addresses.

Use the **delete** form of this command to remove a listen-address.

Use the **show** form of this command to view the listen-address configuration.

service telnet port <port>

Specifies the port the system will use for the Telnet service.

Syntax

set service telnet port *port*

delete service telnet port

show service telnet port

Command Default

The default is port 23.

Parameters

port

The port the system will use for the Telnet service. The range is 1 to 65534.

Modes

Configuration mode

Configuration Statement

```
service {
    telnet {
        port port
    }
}
```

Usage Guidelines

Use this command to specify the port the system will use for the Telnet service.

Use the **set** form of this command to specify the port the system will use for the Telnet service.

Use the **delete** form of this command to restore the default port configuration.

Use the **show** form of this command to view the port configuration.

telnet <address>

Creates a terminal session to a Telnet server.

Syntax

telnet *address*

Parameters

address

Mandatory. The IP address or hostname of the Telnet server to connect to. The system connects through port 23 (the well-known port for the Telnet service).

Modes

Operational mode

Usage Guidelines

Use this command to create a terminal session to a remote machine running a Telnet service.

Examples

The following example shows a telnet session being established to 192.168.1.77.

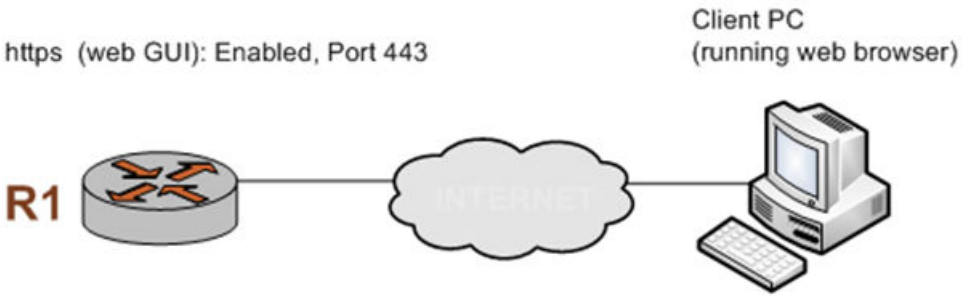
```
vyatta@R1:~$ telnet 192.168.1.77
Entering character mode
Escape character is '^]'.
Welcome to Vyatta
vyatta login:
```

Web GUI Access (HTTPS)

Web GUI access configuration

Configuring web GUI access is optional, but creating the HTTPS service will allow you to access the web GUI on the Brocade vRouter remotely via a web browser. The following table enables HTTPS on the default port (port 443), as shown in the following figure.

FIGURE 3 Enabling web GUI access



To enable the HTTPS service on the Brocade vRouter to provide access to the web GUI, perform the following steps in configuration mode.

TABLE 3 Enabling web GUI access

Step	Command
Create the configuration node for the HTTPS service.	`vyatta@R1# set service https`
Commit the information.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show service`
	` https {`
	` }`

Web GUI Access Commands

service https

Configures access to the web GUI.

Syntax

set service https

delete service https

show service https

Modes

Configuration mode

Configuration Statement

```
service {
    https
}
```

Usage Guidelines

Use this command to configure access to the web GUI via HTTPS (port 443). Once configured, the web GUI can be accessed by specifying one of the system IP addresses from a web browser.

When the HTTPS service is enabled, HTTP redirection is also enabled. If you want to disable HTTP redirection, use the **service https http-redirect** command.

Use the **set** form of this command to create the HTTPS configuration and enable access to the web GUI.

Use the **delete** form of this command to remove the HTTPS configuration. If you delete the HTTPS configuration node you will disable web GUI access to the system.

Use the **show** form of this command to view the HTTPS configuration.

service https http-redirect

Enables or disables HTTP redirection.

Syntax

set service https http-redirect { **enable** | **disable** }

delete service https http-redirect

show service https http-redirect

Command Default

HTTP redirection is enabled.

Parameters

enable

HTTP redirection is enabled.

disable

HTTP redirection is disabled.

Modes

Configuration mode

Configuration Statement

```
service {
    https
        http-redirect [enable|disable]
    }
}
```

Usage Guidelines

Use this command to specify whether to enable or disable HTTP redirection.

When the Brocade vRouter web GUI service (https) is enabled, HTTP redirection is enabled automatically. This allows the user to specify an HTTP URL rather than an HTTPS URL to connect to the web GUI. In certain circumstances this behavior is undesirable. Use this command to disable this behavior.

Use the **set** form of this command to enable or disable HTTP redirection.

Use the **delete** form of this command to restore the default behavior for HTTP redirection.

Use the **show** form of this command to view HTTP redirection configuration.

service https listen-address <ipv4>

Configures access to the web GUI on a specific address.

Syntax

set service https listen-address *ipv4*

delete service https listen-address *ipv4*

show service https listen-address

Command Default

Requests to access the web GUI will be accepted on any system IP address.

Parameters

ipv4

Multi-node. An IP address that the HTTPS service listens for connection requests on. The address must be assigned to an interface.
You can define more than one **listen-address** by creating multiple **listen-address** configuration nodes.

Modes

Configuration mode

Configuration Statement

```
service {
    https {
        listen-address ipv4
    }
}
```

Usage Guidelines

Use this command to configure the system to accept requests for web GUI access on specific addresses. This provides a way to limit access to the system.

Use the **set** form of this command to configure the system to accept requests for web GUI access on specific addresses.

Use the **delete** form of this command to remove a listen-address.

Use the **show** form of this command to view the listen-address configuration.

restart https

Restarts the HTTPS server.

Syntax

restart https

Modes

Operational mode

Usage Guidelines

Use this command to restart the HTTPS server.

Examples

The following example shows the output resulting from the **restart https** command.

```
vyatta@R1> restart https
Stopping web server: lighttpd.
Starting web server: lighttpd.
Stopping PAGER server
Starting PAGER server
Stopping API PAGER server
Starting API PAGER server
spawn-fcgi: child spawned successfully: PID: 4219
vyatta@R1>
```

NETCONF

NETCONF Overview

NETCONF is a protocol that provides mechanisms for installing, manipulating, and deleting the configuration of network devices. It uses an Extensible Markup Language (XML)-based data encodingfor the configuration data and the protocol messages. The NETCONF operations are realized as remote procedure calls (RPCs).

Refer to RFC 6241, *Network Configuration Protocol (NETCONF)* at https://tools.ietf.org/html/rfc6241 for more information.

NETCONF on the Brocade vRouter

On the Brocade vRouter, NETCONF is used within a Secure Shell (SSH) session through the SSH connection protocol. This mapping allows NETCONF to be run from a secure shell session by a user or an application. This mapping also makes sure that NETCONF complies with SSH IPv6.

On the Brocade vRouter, NETCONF is intended as a machine interface for management software and not intended as a user interface.

Refer to RFC 6242, *Using the NETCONF Protocol over Secure Shell (SSH)* at https://tools.ietf.org/html/rfc6242 for more information on using the NETCONF configuration protocol over SSH.

NETCONF Capabilities Supported on the Brocade vRouter

A NETCONF capability is a set of functions that supplements the base NETCONF specification. The capability is identified by a uniform resource identifier (URI). Capabilities augment the base operations of the device, describing both additional operations and the content that is allowed inside the operations. The client discovers the capabilities of the server and uses any additional operations, parameters, and content that are defined by those capabilities.

Following are the NETCONF capabilities that are supported on the Brocade vRouter:

- capability:candidate 1.0
- capability:startup 1.0
- capability:rollback-on-error 1.0
- capability:validate 1.1

Refer to RFC 6241, *Network Configuration Protocol (NETCONF)* at https://tools.ietf.org/html/rfc6241 for more information on these capabilities.

Initiating a NETCONF Session

To initiate a NETCONF session on the Brocade vRouter, use the commands that are shown in the following table.

TABLE 4 Table 4-1 Initiating a NETCONF session

Step	Command
Initiate NETCONF.	`vyatta@R1# set service netconf`
Define the port for NETCONF over SSH.	`vyatta@R1# set service ssh port 830`
Commit the change.	`vyatta@R1# commit`

YANG Model for NETCONF Monitoring

The **<get-schema>** operation is supported on the Brocade vRouter to query and retrieve schema information and NETCONF state information from a NETCONF server.

Refer to RFC 6022, *YANG Module for NETCONF Monitoring* at https://tools.ietf.org/html/rfc6022 for more information on using **<get-schema>**.

NETCONF Commands

ping

The **ping** command displays whether a destination responded and how long the destination took to receive a reply. If there is an error in the delivery to the destination, the command displays an error message.

TABLE 5 Information about the ping command

Parameter	Description
Sample XML program	`<ping xmlns="urn:vyatta.com:mgmt:vyatta-op"> <host>127.0.0.1</host> <count>5</count> <ttl>3</ttl> </ping>`
XML tags	• host: IP address you want to ping. • count: Number of packets with which you are pinging. • ttl: Time to live(ttl) in an IP packet in seconds that tells a network router whether the packet has been in the network too long and should be discarded. By default, the TTL value is 255.
Sample rpc-reply	`<tx-packet-count>5</tx-packet-count><rx-packe t-count>5</rx-packet-count><min-delay>54</ min-delay><average-delay>62</average-delay><max-de lay>74</max-delay>netconf>`

interface

The **interface** command displays information about an interface name. The command output displays all the IP addresses that are associated with the interface, administrator status, operational status, and description of the interface.

TABLE 6 Information about the interface command

Parameter	Description
Sample XML program	`<interface xmlns="urn:vyatta.com:mgmt:vyatta-op"> <name>br0</name> </interface>`
XML tags	name: Name of an interface
Sample rpc-reply	`<address> <ip>15.15.15.15/24</ip> </address> <address> <ip>2020::/64</ip> </address> <address> <ip>2001::15/64</ip> </address> <address> <ip>3001::211:22ff:fe33:4455/64</ip> </address> <admin-status>up</admin-status> <oper-status>up</oper-status> <description>sample bridge</description>`

route

The **route** command displays information about the path taken to a particular destination address.

TABLE 7 Information about the route command

Parameter	Description

Sample XML program	`<route xmlns="urn:vyatta.com:mgmt:vyatta-op">`
	`<destination>192.168.14.0/24</destination>`
	`</route>`

XML tags	• destination (optional): IP address or IP prefix • family: ipv4 (default) or ipv6 NOTE When the destination is not present, the entire route table for the specified family is returned as the output.

Sample rpc-reply	`<route>`
	`<destination>192.168.14.0</destination>`
	`<path>`
	`<entry>1</entry>`
	`<nexthop>31.31.31.32</nexthop>`
	`<device>dp0p256p1</device>`
	`</path>`
	`</route>`

SNMP

SNMP overview

SNMP (Simple Network Management Protocol) is a mechanism for managing network and computer devices.

SNMP uses a manager/agent model for managing the devices. The agent resides in the device and provides the interface to the physical device being managed. The manager resides on the management system and provides the interface between the user and the SNMP agent. The interface between the SNMP manager and the SNMP agent uses a Management Information Base (MIB) and a small set of commands to exchange information.

The Brocade vRouter supports SNMP over both IPv4 and IPv6 networks.

The following list describes the SNMP components.

- MIB objects—A MIB contains the set of variables and objects that are managed (for example, MTU on a network interface). The objects are organized into a tree structure in which each object is a leaf node. Each object has its unique Object Identifier (OID). Objects are of two types: *scalar* and *tabular*. A scalar object defines a single object instance. A tabular object defines multiple related object instances that are grouped in MIB tables. For example, the uptime on a device is a scalar object, but the routing table in a system is a tabular object.
- Traps—In addition to MIB objects, the SNMP agent on a device can formulate alarms and notifications into SNMP traps. The device asynchronously sends the traps to the SNMP managers that are configured as trap destinations or targets. This sending of traps keeps the network manager informed of the status and health of the device. Traps are unacknowledged by the remote application that receives the message. The Brocade vRouter uses User Datagram Protocol (UDP) for traps. For SNMP requests, UDP port 161 is used. For SNMP traps, UDP port 162 is used. SNMPv2 and SNMPv3 support traps. Traps can be configured for each routing protocol.

  NOTE
  Protocols BFD, BGP, and OSPF are supported for SNMP traps and these traps are disabled by default.

- Informs— Informs are acknowledged traps. After receiving an inform notification, a remote application sends back an acknowledge message indicating that it received the message. By default, the Brocade vRouter uses UDP for inform notifications and sends inform notifications to trap targets.

  NOTE
  SNMPv3 supports informs.

SNMP commands

SNMP commands can be used to read or change configuration or to perform actions on a device, such as resetting it. The set of commands used in SNMP are: **GET**, **GET-NEXT**, **GET-RESPONSE**, **SET**, and **TRAP**.

- **GET** and **GET-NEXT** are used by the SNMP manager to request information about an object. These commands are used to view configuration or status or to poll information, such as statistics.
- **SET** is used by the SNMP manager to change the value of a specific object. Setting a configuration object changes the configuration of the device. Setting an executable object performs an action, such as a file operation or a reset.
- **GET-RESPONSE** is used by the SNMP agent on the device to return the requested information by **GET** or **GET-NEXT** or the status of the **SET** operation.
- The **TRAP** command is used by the agent to asynchronously inform the manager about events important to the manager.

SNMP versions

Currently, SNMP has three versions:

- SNMPv1—This version is the first version of the protocol. It is described in RFC 1157.
- SNMPv2—This version is an evolution of the first version, and it adds a number of improvements to SNMPv1. It is described in RFCs 1902 through 1908.
- SNMPv3—This version improves the security model in SNMPv2 and adds support for proxies. It is described in RFCs 3413 through 3415.

The Brocade vRouter supports SNMPv2 with community string (SNMPv2c) and SNMPv3 with SNMPv3 users.

SNMPv3

SNMPv3 adds security features to address the security shortcomings of SNMPv1 and SNMPv2. For information standards for SNMPv3 that are supported on the Brocade vRouter, see Supported standards on page 48.

The SNMPv3 architecture uses a modular approach to allow the protocol to be adapted in the future, if and when other types of features are added. The architecture supports the simultaneous use of different security, access control, and message processing models.

The SNMPv3 architecture provides the following security-related models:

- User-based Security Model (USM)—Used for message security. This model is defined in RFC 3414.
- Transport Security Model (TSM)—Used for message security. This model is defined in RFC 5591.
- View-based Access Control Model (VACM)—Used for access control. This model is defined in RFC 2275.

The Brocade vRouter currently supports all three models.

The SNMPv3 architecture supports the following security features through USM and TSM:

- Data integrity—Ensures that packets have not been altered or destroyed in transit.
- Data-origin authentication—Verifies that the received packets come from a valid source.
- Data confidentiality— Encrypts packets to prevent data from being disclosed to unauthorized sources.
- Message timeliness and replay protection— Ensures a packet whose generation time is outside of a specified time window is not accepted.

USM

The User-based Security Model (USM) provides SNMP message-level security and is the default security model for SNMPv3. It also uses the traditional concept of a user (identified by a username) with which to associate security information. This model uses UDP to send the SNMP packets.

The following table lists the security protocols and modules used in the USM model to provide the SNMP message-level security.

TABLE 8

Module	Function	Notes
Authentication	Provides for data integrity and data-origin authentication	The following authentication protocols are supported: • HMAC-MD5-96 • HMAC-SHA-96 The entire message is checked for integrity. For a message to be authenticated, it needs to pass the authentication check and the timeliness check.
Privacy	Provides data confidentiality	The following encryption protocols are supported to encrypt messages: • Advanced Encryption Standard (AES) • Data Encryption Standard (DES) Note: If privacy is used, then the message also requires authentication.
Timeliness	Provides message timeliness and replay protection	The timeliness values in an SNMP message are used to do timeliness checking. This checking is performed only if authentication is applied to the message.

To authenticate or encrypt, or authenticate and encrypt the messages between an SNMP manager and an SNMP agent, the SNMP pair must share secret keys—an authentication secret key for authentication and an encryption secret key for encryption. Before using SNMPv3, you must first configure the secret keys so that they are added to the databases of the SNMP managers and agents that are to share the keys.

TSM

The Transport Security Model (TSM) within the SNMPv3 architecture is designed for use with secure transport protocols, such as SNMP over Secure Shell (SSH), Transport Layer Security (TLS), or Datagram Transport Layer Security (DTLS) to send SNMPv3 packets through secure tunnels. Brocade vRouter supports TLS and DTLS in its SNMPv3 implementation.

> NOTE
> The current implementation of TSM does not support SNMP over SSH.

TLS and DTLS use X.509 certificates to authenticate both the client and server of the secure tunnel connections. A public key infrastructure (PKI) is required to generate these certificates. To employ TLS and DTLS, you are required to generate X.509 security keys and certificates and install them on both the SNMP manager and the SNMP agent. The generation and distribution of certificates and keys using PKI involves numerous complex security issues, which are outside the scope of this document. Consult your particular PKI deployment documentation for the necessary procedures to generate and distribute these certificates and keys.

VACM

The View-based Access Control Model (VACM) is used for access control. In this model, access control is determined based on V3 groups and community. A group defines the access policy or the read-and-write access privileges for a set of SNMPv3 users. A group also defines the type of MIB view provided to a set of users. A group defines the following:

- Which users are allowed to access which view (a MIB or MIB object within a MIB)
- What type of access privileges are allowed into a view

NOTE
The Brocade vRouter supports the access privilege types of read-only (ro) and read-write (rw) for groups.

Choosing USM or TSM

With two security models available, how do you determine which model to use in your network environment?

The main advantage of using TSM is the ability to integrate SNMP management into the existing X.509 public key security infrastructure of an organization.

Consider implementing TSM if you already have an X.509 public key infrastructure, need to deploy an X.509 public key infrastructure, or do not have a system for managing USM private keys in SNMPv3.

Consider implementing USM if you do not need to deploy an X.509 public key infrastructure or you already have a system for managing USM private keys for use in SNMPv3.

Default object IDs

Two default object IDs set by Brocade vRouter are as follows:

- sysObjectID = 1.3.6.1.4.1.30803
- sysDescr = Vyatta VSE6.6R0S6

The sysDescr object ID is updated automatically with each new release. It can also can be changed by using the **service snmp description** *desc* command.

Supported standards

The Brocade vRouter implementation of SNMPv1, SNMPv2, and SNMPv3 complies with the following standards:

- RFC 1525, *Definitions of Managed Objects for Source Routing Bridges*
- RFC 2742, *Definitions of Managed Objects for Extensible SNMP Agents*
- RFC 2786, *Diffie-Helman USM Key Management Information Base and Textual Convention*
- RFC 2856, *Textual Conventions for Additional High Capacity Data Types*
- RFC 2864, *The Inverted Stack Table Extension to the Interfaces Group MIB*
- RFC 3165, *Definitions of Managed Objects for the Delegation of Management Scripts*
- RFC 3231, *Definitions of Managed Objects for Scheduling Management Operations*
- RFC 3411, *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*
- RFC 3412, *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*
- RFC 3413, *Simple Network Management Protocol (SNMP) Applications*
- RFC 3414, *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*

- RFC 3415, *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*
- RFC 3417, *Transport Mappings for the Simple Network Management Protocol (SNMP)*
- RFC 3419, *Textual Conventions for Transport Addresses*
- RFC 3584, *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*
- RFC 3635, *Definitions of Managed Objects for the Ethernet-like Interface Types*
- RFC 3826, *The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model*
- RFC 4001, *Textual Conventions for Internet Network Addresses*
- RFC 4273, *Definitions of Managed Objects for BGP-4*
- RFC 5591, *Transport Security Model for the Simple Network Management Protocol (SNMP)*
- RFC 5953, *Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP)*

Supported MIBs

MIBs are typically located in the `/usr/share/snmp/mibs` directory.

The following table lists the standard MIBs and traps supported by the Brocade vRouter. RFCs can be found at http://tools.ietf.org.

TABLE 9 Supported standard MIBs

MIB Name	Document Title	OIDs	Notes
BFD-MIB	RFC 7331, *Bidirectional Forwarding Detection (BFD) Management Information Base*	1.3.6.1.2.1.222	Version 22 of the BFD protocol MIB is supported, except the Echo Packet and Drop Counters. The following traps are supported: • bfdSessDiag • bfdSessDown • bfdSessUp
BGP4-MIB	RFC 1657, *Definitions of Managed Objects for the Fourth Version of Border Gateway Protocol (BGP-4)*	1.3.6.1.2.1.15	The protocol MIB is supported plus the following traps: • bgpEstablished • bgpBackwardTransition
HOST-RESOURCES-MIB	RFC 2790, *Host Resources MIB*	1.3.6.1.2.1.25	
RMON-MIB	RFC 2819, *Remote Network Monitoring Management Information Base*	1.3.6.1.2.1.16	
IF-MIB	RFC 2863, *The Interfaces Group MIB*	1.3.6.1.2.1.31	The following traps are supported: • linkUp • linkDown
IGMP-MIB	RFC2933, *Internet Group Management Protocol MIB*	1.3.6.1.2.1.85	
EVENT-MIB	RFC 2981, *Event MIB*		
IP-MIB	RFC 2011, *SNMPv2 Management Information Base for the Internet Protocol using SMIv2*	1.3.6.1.2.1.48	Only packets in which the Brocade vRouter is an endpoint are accounted. Forwarded traffic is not accounted.
NOTIFICATION-LOG-MIB	RFC 3014, *Notification Log MIB*	1.3.6.1.2.1.92	
IPv6-MLD-MIB	RFC 3019, *IP Version 6 Management Information Base for*	1.3.6.1.2.1.91	

TABLE 9 Supported standard MIBs (continued)

MIB Name	Document Title	OIDs	Notes
	The Multicast Listener Discovery Protocol		
IPM-ROUTE	RFC 2932, *IPv4 Multicast Routing MIB*	1.3.6.1.2.1.83	
IPV6-UDP-MIB	RFC 2454, *IP Version 6 Management Information Base for the User Datagram Protocol*	1.3.6.1.2.1.7	Only packets in which the Brocade vRouter is an endpoint are accounted. Forwarded traffic is not accounted.
KEEPALIVED-MIB	Authored by Vincent Bernat. Extends the keepalived daemon to support the Net-SNMP agentx protocol. Provides additional information specific to the Brocade vRouter implementation, such as state information, sync group state information, and so on.	1.3.6.1.4.1.9586.100.5	
NAT-MIB	RFC 4008, *Definitions of Managed Objects for Network Address Translators (NAT)*	1.3.6.1.2.1.123	
PIM-MIB	RFC 2934, *Protocol Independent Multicast MIB for IPv4*	1.3.6.1.3.61	
RFC1213-MIB	RFC 1213, *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*	1.3.6.1.2.1	
RIPv2-MIB	RFC 1724, *RIP Version 2 MIB Extension*	1.3.6.1.2.1.23	
SNMPv2-MIB	RFC 3418, *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*	1.3.6.1.6.3.1	The following traps are supported: • coldStart • warmStart
TCP-MIB	RFC 4022, *Management Information Base for the Transmission Control Protocol (TCP)*	1.3.6.1.2.1.49	Only packets in which the Brocade vRouter is an endpoint are accounted. Forwarded traffic is not accounted.
UDP-MIB	RFC 4113, *Management Information Base for the User Datagram Protocol (UDP)*	1.3.6.1.2.1.50	Only packets in which the Brocade vRouter is an endpoint are accounted. Forwarded traffic is not accounted.
IP-Forward-MIB	RFC 4292, *IP Forwarding Table MIB*	1.3.6.1.2.1.4.24	
OSPF-MIB	RFC 4750, *OSPF Version 2 Management Information Base*	1.3.6.1.2.1.14	The following OSPF traps are supported: • ospfTrapControl • ospfTraps • ospfTrapConformance
LLDPD-MIB	no RFC		The MIB module defines objects for Linux implementation of IEEE 802.1ab Link Layer Discovery Protocol (LLDP).

TABLE 9 Supported standard MIBs (continued)

MIB Name	Document Title	OIDs	Notes
UCD-DISKIO-MIB	no RFC	1.3.6.1.4.1.2021.13.15.1	A table of IO devices and how much data they have read and written.

SNMP configuration examples

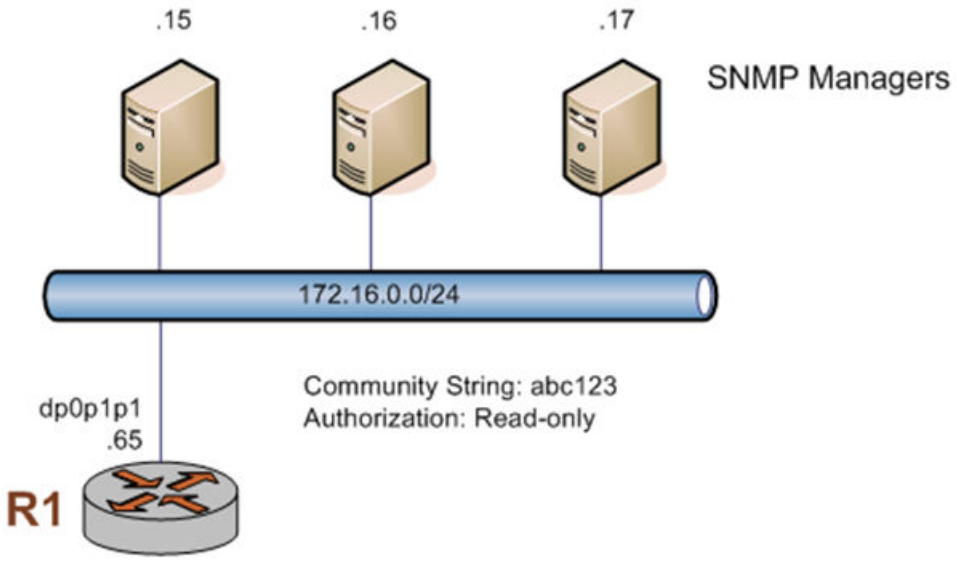
This section presents the following topics:

- Defining the SNMP community
- Assigning views to an SNMP community
- Specifying protocol-specific SNMP traps
- Specifying trap destinations
- SNMP over IPv6

To configure SNMP, the Brocade vRouter MIB model must be loaded.

At the end of running these configuration procedures, you set up an SNMP community that includes three hosts, which serves as SNMP managers, and configures the system R1 to send traps to all the three managers. When you have finished, the system is configured as shown in the following figure.

FIGURE 4 Configuring SNMP communities and traps



Defining the SNMP community

SNMP community strings are used only by systems that support SNMPv1 and SNMPv2c protocols. SNMPv3 uses a username and password authentication, along with an encryption key.

The SNMP community of a system is the list of SNMP clients authorized to make requests of the system. Authorization for the community is in the form of a community string. The community string acts as a password, providing basic security and protecting the system against spurious SNMP requests.

- If no SNMP clients or networks are explicitly defined, then any client presenting the correct community string is granted the access privilege specified in the **authorization** option.
- If any client or network is defined, then only explicitly listed clients or networks are granted access to the system. Those clients have the access privilege specified by the **authorization** option. (The default is read-only.)

With reference to the figure **Configuring SNMP communities and traps**, the following configuration example shows how to set the SNMP community string for the system R1 to `abc123` and specify three clients for the community with the following IP addresses: `176.16.0.15`, `176.16.0.16`, and `176.16.0.17`. Read-only access is provided for this community.

TABLE 10 Defining an SNMP community

Step	Command
Create the **snmp** configuration node and the **community** configuration node. Set the community string. Note that using the **edit** command creates the community if it does not already exist. Navigate to the configuration node of the community for easier configuration.	`vyatta@R1# edit service snmp community abc123` `[edit service snmp community abc123]`

List the SNMP clients making up this community.	`vyatta@R1# set client 176.16.0.15`
	`vyatta@R1# set client 176.16.0.16`
	`vyatta@R1# set client 176.16.0.17`

Set the privilege level for this community to read-only.	`vyatta@R1# set authorization ro`
Commit the change.	`vyatta@R1# commit`

Verify the configuration.	`vyatta@R1# show`
	` authorization ro`
	` client 176.16.0.15`
	` client 176.16.0.16`
	` client 176.16.0.17`

Return to the top of the configuration tree.	`vyatta@R1# top`

Assigning views to an SNMP community

After you define an SNMP community and a view, you can associate each community with any number of views.

With reference to the figure **Configuring SNMP communities and traps**, the following example shows how to add views to a community.

TABLE 11 Assigning Views to an SNMP Community

Step	Command
Specify a sub-tree to appear in the view.	`vyatta@R1# set service snmp view myview oid` `1.3.6.1.2.1.4`

TABLE 11 Assigning Views to an SNMP Community (continued)

Step	Command
Associate the view with the community.	`vyatta@R1# set service snmp community abc123 view myview`
Commit the changes.	`vyatta@R1# commit`

Display the configuration.	`vyatta@R1# show service snmp`
	`snmp {`
	`community abc123 {`
	`view myview`
	`}`
	`view myview {`
	`oid 1.3.6.1.2.1.4`
	`}`
	`}`

Specifying protocol-specific traps

Brocade vRouter supports specifying SNMP traps for each routing protocol in your configuration. Currently, this feature is supported for BGP, BFD, and OSPF protocols.

> **NOTE**
> SNMP traps for all protocols are disabled by default on the Brocade vRouter system. You must enable the SNMP traps by using the **set service snmp notification** command.

All notifications as defined in the supported MIBs for each routing protocol are enabled by using the keyword **all** in the command syntax.

In the following figure, the community string for R1 is defined as `abc123`. The traps from R1 are configured to be sent to the three SNMP Managers that are defined by the IP addresses of `176.16.0.15`, `176.16.0.16`, and `176.16.0.17`, respectively. R1 is connected to R2 and R3 by using a routing protocol, such as BFD. The following configuration example shows how to specify BFD-specific SNMP traps for R1.

> **NOTE**
> The command to enable protocol-specific SNMP traps is similar for all routing protocols. Refer to the SNMP commands section for more information on the command syntax.

FIGURE 5 Specifying protocol-specific traps

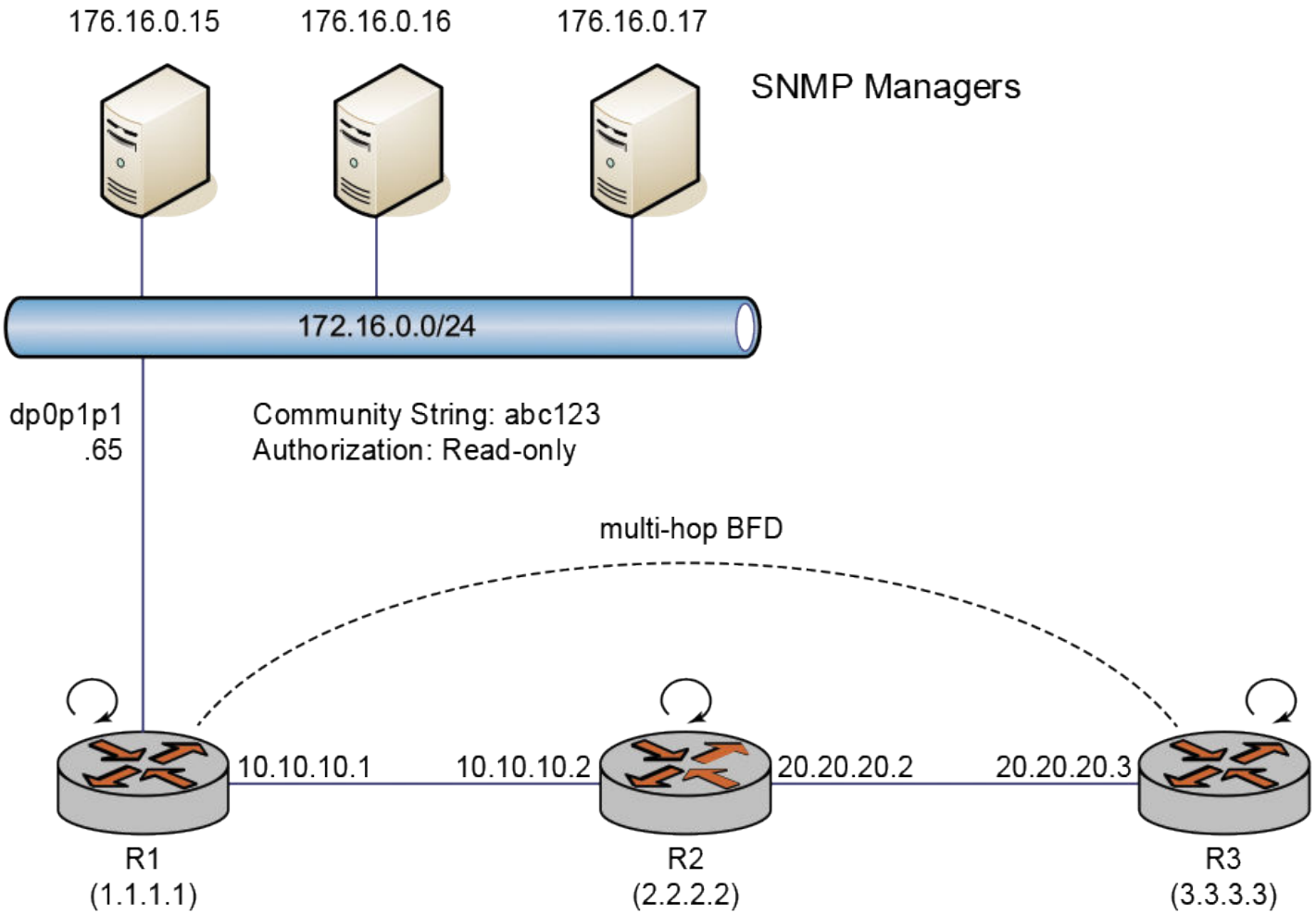


TABLE 12 Specifying protocol-specific traps

Step	Command
Enable all SNMP traps for BFD on R1.	`vyatta@R1# set service snmp notification bfd all`
Commit the change.	`vyatta@R1# commit`

View the change.	`vyatta@R1# show bfd`
	`BFD ID: 00 Start Time:Thu Jan 1 00:00:16 1970`
	`BFD Admin State: DOWN`
	`Number of Sessions: 0`
	`Slow Timer: 2000 Image type: MONOLITHIC`
	`Echo Mode: Disabled BFD Notifications enabled`
	`Next Session Discriminator: 1`
	
	`vyatta@R1# show service snmp`
	` service {`
	`  snmp {`
	`  community abc123 {`

TABLE 12 Specifying protocol-specific traps (continued)

Step	Command

	authorization ro
	}
	notification {
	bfd {
	all
	}
	
	}
	}
	ssh
	}

Specifying trap destinations

After you specify an SNMP trap for a particular protocol, you must specify the SNMP trap destination as one or some of the configured SNMP managers.

With reference to the figure **Configuring SNMP communities and traps**, the following configuration example shows how to direct the system R1 to send SNMP traps to the configured network managers at `176.16.0.15`, `176.16.0.16`, and `176.16.0.17`.

TABLE 13 Specifying SNMP trap destinations

Step	Command

Define the trap destinations, one at a time.	`vyatta@R1# set service snmp trap-target 176.16.0.15`
	`vyatta@R1# set service snmp trap-target 176.16.0.16`
	`vyatta@R1# set service snmp trap-target 176.16.0.17`

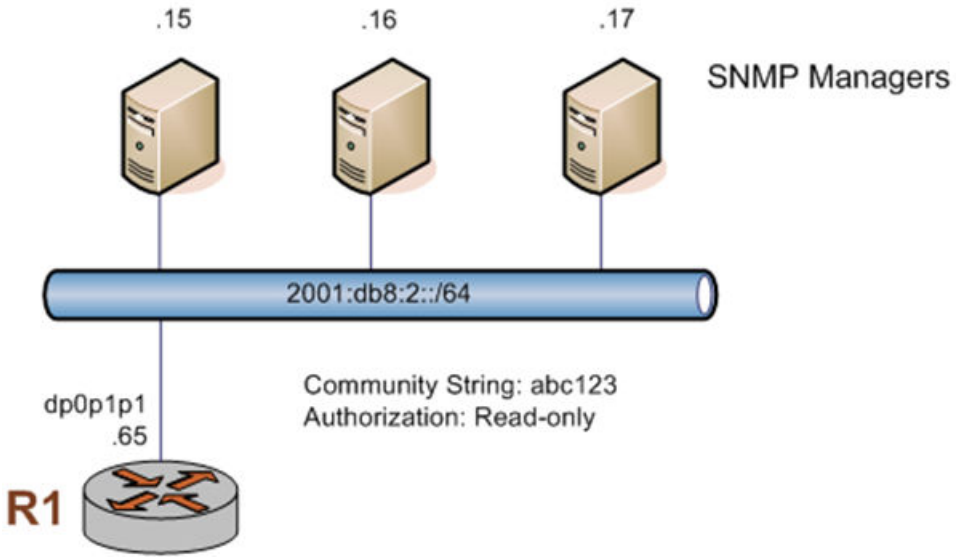
Commit the change.	`vyatta@R1# commit`

Verify the configuration.	`vyatta@R1# show service snmp trap-target`
	` trap-target 176.16.0.15 {`
	` }`
	` trap-target 176.16.0.16 {`
	` }`
	` trap-target 176.16.0.17 {`
	` }`

SNMP over IPv6

This sequence is the same as the previous example but uses IPv6 addresses. When you have finished, the system is configured as shown in the following figure.

FIGURE 6 Configuring SNMP communities and traps - IPv6



To define the SNMP configuration, perform the following steps in configuration mode.

TABLE 14 Defining the SNMP configuration

Step	Command
Create the **snmp** configuration node and the **community** configuration node. Set the community string.	`vyatta@R1# set service snmp community abc123`

List the SNMP clients making up this community.	`vyatta@R1# set service snmp community abc123 client 2001:db8:2::15`
	`vyatta@R1# set service snmp community abc123 client 2001:db8:2::16`
	`vyatta@R1# set service snmp community abc123 client 2001:db8:2::17`

Set the privilege level for this community to read-only.	`vyatta@R1# set service snmp community abc123 authorization ro`

Define the trap destinations, one at a time.	`vyatta@R1# set service snmp trap-target 2001:db8:2::15`
	`vyatta@R1# set service snmp trap-target 2001:db8:2::16`
	`vyatta@R1# set service snmp trap-target 2001:db8:2::17`

Commit the change.	`vyatta@R1# commit`

Verify the configuration.	`vyatta@R1# show service snmp`
	`community abc123 {`
	`authorization ro`
	`client 176.16.0.15`
	`client 176.16.0.16`
	`client 176.16.0.17`
	`client 2001:db8:2::15`
	`client 2001:db8:2::16`

TABLE 14 Defining the SNMP configuration (continued)

Step	Command

	client 2001:db8:2::17
	}
	trap-target 176.16.0.15 {
	}
	trap-target 176.16.0.16 {
	}
	trap-target 176.16.0.17
	}
	trap-target 2001:db8:2::15 {
	}
	trap-target 2001:db8:2::16
	}
	trap-target 2001:db8:2::17 {
	}

SNMPv3 configuration examples

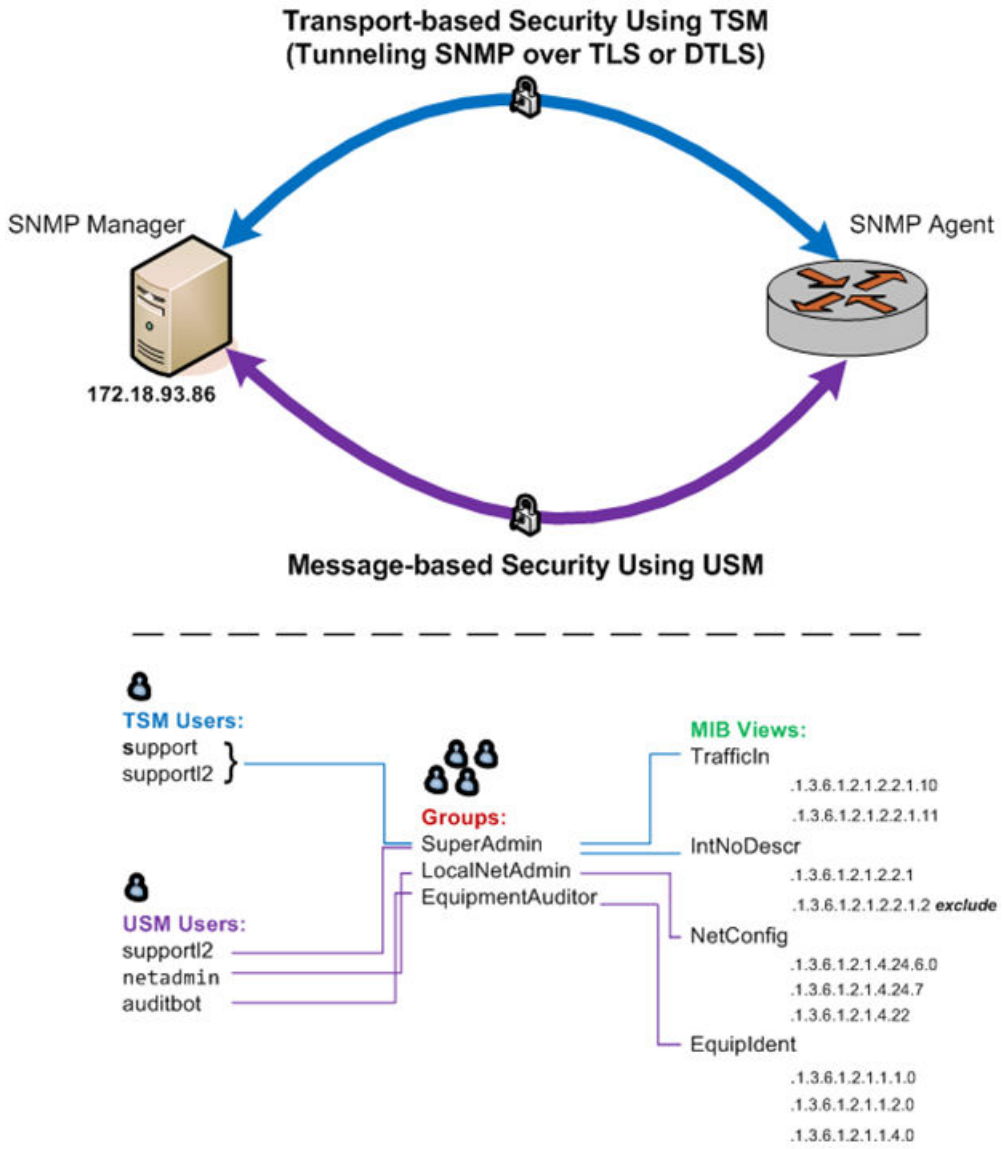
To configure SNMPv3, the Brocade vRouter MIB model must be loaded.

The configurations in this section does the following:

- Defines the SNMPv3 users (USM and TSM users) and SNMPv3 groups
- Assigns the users and views to SNMPv3 groups
- Specifies the destinations to which to send the SNMP trap notifications to the configured SNMP manager at 172.18.93.86

Refer to the following figure for an example of SNMPv3 topology for the configurations in this section.

FIGURE 7 SNMPv3 topology example



Defining the users

This section provides the following topics:

- Defining the USM users
- Defining the TSM users

Defining the USM users

As part of the configuration steps to define the USM users, you are also required to specify the following information:

- Type of security protocol (authentication, privacy, or both) to apply to the SNMP messages sent between an SNMP manager and an SNMP client

- Secret keys associated with the selected security protocols

Before defining the USM users, configure the secret keys associated with the security protocols so that these are added to the databases of the SNMP entities that are to share the keys.

The following table shows the following configurations for USM:

- The auditbot user employs authentication only
- The netadmin and supportl2 users employ authentication and privacy

To define the USM users, perform the following steps in configuration mode. You must specify at least one of the security protocols (authentication or privacy).

NOTE
After defining a user by using the **service snmp v3 user** *username* **auth plaintext-key** *passwd* command and committing the command, a user engine ID (*engineid*) to be associated with the given SNMPv3 user is automatically added to the configuration. The engine ID is used during the generation of an encrypted password based on the configured plain-text password and the validation of passwords.

NOTE
During an upgrade to a new Brocade vRouter image, ensure that you use the same user engine IDs for each of the existing SNMPv3 users.

TABLE 15 Defining the USM users

Step	Command

If you are using the authentication protocol to authenticate the user, specify the name of the user, authentication protocol, and authentication password. In this example, a clear-text password is used to authenticate a user.
NOTE
The clear-text passwords are converted to encrypted keys after the commands are committed.
	`vyatta@R1# set service snmp v3 user auditbot auth plaintext-key auditbotkey`
	`vyatta@R1# set service snmp v3 user netadmin auth plaintext-key netadminkey`
	`vyatta@R1# set service snmp v3 user supportl2 auth plaintext-key supportl2key`

If you are using the privacy protocol to provide data confidentiality for SNMPv3 traffic, specify the name of the user, privacy protocol, and privacy password. In this example, a clear-text password is used to encrypt the SNMP traffic. NOTE The clear-text passwords are converted to encrypted keys after the commands are committed.	`vyatta@R1# set service snmp v3 user netadmin privacy plaintext-key netadminkey1` `vyatta@R1# set service snmp v3 user supportl2 privacy plaintext-key supportl2key1`
Commit the change.	`vyatta@R1# commit`

Verify the configuration.
Note that the clear-text passwords configured for each of the users have been converted to encrypted keys and that engine IDs have been added to each user configuration where the user authentication protocol is used for authenticating the user.
	`vyatta@R1# show service snmp`
	` v3 {`
	`  user auditbot {`
	`  auth {`
	`  encrypted-key 0xba6273b420a64b415ad0a44e80106dbd`
	`  }`
	`  engineid 0x80001f8880141fcc01ca3edd51`
	`  }`
	`  user netadmin {`
	`  auth {`
	`  encrypted-key 0x110c1e3aa857084f9bf7ce4faaf44496`
	`  }`

TABLE 15 Defining the USM users (continued)

Step	Command

	engineid 0x80001f8880141fcc01ca3edd51
	privacy {
	encrypted-key
	0x4d8590f7fb640e35b673443823fccb71
	}
	}
	user support12 {
	auth {
	encrypted-key
	0x9a72fc4e7a3cf01c0eadecb13dcf6f7c
	}
	engineid 0x80001f8880141fcc01ca3edd51
	privacy {
	encrypted-key
	0x792dedb243b0fcbb7662b802f1444671
	}
	}
	}

Verify the configuration.	vyatta@R1:~$ show snmp v3 user SNMPv3 Users: User Auth Priv Mode Group ---- ---- ---- ---- ----- auditbot md5 ro netadmin md5 des ro support12 md5 des ro

Defining the TSM users

When defining a TSM user, you are also required to specify the TSM certificate of the user (either the certificate fingerprint or the file that holds the certificate). During the user configuration, also specify the TSM certificate of the SNMP agent (either the certificate fingerprint or the file that holds the certificate).

Before configuring a TSM user, create the X.509 user keys and certificates for the associated SNMP manager and agent, and then install each key-and-certificate pair on the paired SNMP entities.

> **NOTE**
> The generation and distribution of certificates and keys by using PKI involves numerous complex security issues, which are outside the scope of this document. Consult your particular PKI deployment documentation for the necessary procedures to generate and distribute these certificates and keys.

> **NOTE**
> The location of the certificates and keys on the SNMP-manager system is dependent on the specific SNMP management software used.

Perform the following steps before configuring TSM:

1. Generate the X.509 user key and certificate (one pair) for each of the paired SNMP entities.
2. Add the security keys for the SNMP agent and SNMP manager to the `~/.snmp/tls/private/` directory.
3. Add the certificates for the SNMP agent and SNMP manager to the `~/.snmp/tls/certs/` directory.

TSM configuration example

The example shows the following configurations for TSM:.

- TSM user support and the file support.crt that holds the TSM certificate of this user
- TSM user supportl2 and the file supportl2.crt that holds the TSM certificate of this user

  NOTE
  The TSM user supportl2 is also configured as a USM user. See Defining the USM users on page 58

- File `snmpd.crt` that holds the TSM certificate of the SNMP agent

To define the TSM users and specify the TSM certificates for the TSM users and an SNMP agent, perform the following steps in configuration mode.

TABLE 16 Defining the TSM Users and Specifying TSM Certificates for TSM Users and an SNMP Agent

Step	Command
Specify the name of the TSM user and TSM certificate of the user (either the certificate fingerprint or the file that holds the certificate).	`vyatta@R1# set service snmp v3 user support tsm-key support.crt` `vyatta@R1# set service snmp v3 user supportl2 tsm-key supportl2.crt`
Specify the TSM certificate of the SNMP agent (either the certificate fingerprint or file that holds the certificate).	`vyatta@R1#set service snmp v3 tsm local-key snmpd.crt`
Commit the change.	`vyatta@R1# commit`

Verify the configuration.	`vyatta@R1:~$show snmp v3 user`
	
	`SNMPv3 Users:`
	
	`User Auth Priv Mode Group`
	`---- ---- ---- ----- -----`
	`auditbot md5 ro`
	`netadmin md5 des ro`
	`support ro`
	`supportl2 md5 des ro`

Verify the configuration.	`vyatta@R1:~$ show snmp v3 certificates`
	`/etc/snmp/tls:`
	
	`certs/snmpd.crt:`
	`subject= /C=US/ST=CA/L=Davis/O=Net-SNMP/`
	`OU=Development/CN=raji/`
	`emailAddress=raji.mti@vyatta.com`
	`SHA1 Fingerprint=DB:C8:BC:07:40:0D:A6:68:EF:7D:`
	`3E:CB:1B:22:52:E5:FA:FE:3D:D3`
	
	`certs/support.crt:`
	`subject= /C=US/ST=CA/L=Davis/O=Vyatta/`
	`OU=Development/CN=raji/`
	`emailAddress=raji.mti@vyatta.com`
	`SHA1 Fingerprint=A9:E7:17:31:2A:84:96:DE:19:EE:2D:`
	`36:D8:FD:B1:97:F9:A3:FF:1B`
	
	`certs/supportl2.crt:`
	`subject= /C=US/ST=CA/L=Davis/O=Vyatta/`
	`OU=Development/CN=raji/`
	`emailAddress=raji.mti@vyatta.com`
	`SHA1 Fingerprint=E9:4B:07:26:8E:65:C2:EC:`
	`25:37:76:15:9C:12:DC:EF:FA:FA:81:04`

Defining MIB views

To define a MIB view, specify the name of the view and the SNMP Object Identifier (OID) of the subtree to be included or excluded in the view. To identify a single row in a MIB table to be included within the view, specify a bit mask.

The following table shows the MIB views named EquipIdent, NetConfig, TrafficIn, and IntNoDescr and the OID subtrees to be included or excluded in these views.

To define MIB views, perform the following steps in configuration mode.

TABLE 17 Defining MIB views

Step	Command

Specify a name for a MIB view and define an OID subtree to be included in the view. Configure each MIB view one at a time.	`vyatta@R1# set service snmpview EquipIdent oid 1.3.6.1.2.1.1.1.0`
	`vyatta@R1# set service snmpview EquipIdent oid 1.3.6.1.2.1.1.2.0`
	`vyatta@R1# set service snmpview EquipIdent oid 1.3.6.1.2.1.1.4.0`
	`vyatta@R1# set service snmpview NetConfig oid 1.3.6.1.2.1.4.24.6.0`
	`vyatta@R1# set service snmpview NetConfig oid 1.3.6.1.2.1.4.24.7`
	`vyatta@R1# set service snmpview NetConfig oid 1.3.6.1.2.1.4.22`
	`vyatta@R1# set service snmpview TrafficIn oid 1.3.6.1.2.1.2.2.1.10`
	`vyatta@R1# set service snmpview TrafficIn oid 1.3.6.1.2.1.2.2.1.11`
	`vyatta@R1# set service snmpview IntNoDescr oid 1.3.6.1.2.1.2.2.1`

Specify a name for the MIB view and define an OID subtree to be excluded from the view.	`vyatta@R1# set service snmpview IntNoDescr oid 1.3.6.1.2.1.2.2.1.2 exclude`
Commit the change.	`vyatta@R1# commit`

Verify the configuration.	`vyatta@R1:~$ show snmpview`
	`SNMP Views:`
	`View : EquipIdent`
	`OIDs :`
	`.1.3.6.1.2.1.1.1.0`
	`.1.3.6.1.2.1.1.2.0`
	`.1.3.6.1.2.1.1.4.0`
	`View : IntNoDescr`
	`OIDs :`
	`.1.3.6.1.2.1.2.2.1`
	`.1.3.6.1.2.1.2.2.1.2 exclude`
	`View : NetConfig`
	`OIDs :`
	`.1.3.6.1.2.1.4.22`
	`.1.3.6.1.2.1.4.24.6.0`
	`.1.3.6.1.2.1.4.24.7`
	`View : TrafficIn`
	`OIDs :`
	`.1.3.6.1.2.1.2.2.1.10`
	`.1.3.6.1.2.1.2.2.1.11`

Defining user groups and assigning users and views to groups

To define an SNMPv3 user group, specify the name of the group. By default, the Brocade vRouter supports the access privilege type of read-only (ro) for user groups. You do not need to set this parameter explicitly when defining a user group. After defining user groups, assign the configured users and views to them.

> NOTE
> Currently, the Brocade vRouter only read-only privileges. It does not support read-write privileges.

The following table shows the following configurations:

- The user groups named EquipmentAuditor, LocalNetAdmin, and SuperAdmin
- Assignment of the users auditbot, netadmin, supportl2, and support to a user group
- Assignment of the views EquipIdent, NetConfig, TrafficIn, and IntNoDescr to a user group

To define user groups and assign users and views to the groups, perform the following steps in configuration mode.

TABLE 18 Defining user groups and assigning users and views to user groups

Step	Command

Define an SNMPv3 user group, one user at a time.	`vyatta@R1# set service snmp v3 group EquipmentAuditor`
	`vyatta@R1# set service snmp v3 group LocalNetAdmin`
	`vyatta@R1# set service snmp v3 group SuperAdmin`

Assign users to an SNMPv3 user group, one user at a time.	`vyatta@R1# set service snmp v3 user auditbot group EquipmentAuditor`
	`vyatta@R1# set service snmp v3 user netadmin group LocalNetAdmin`
	`vyatta@R1# set service snmp v3 user supportl2 group SuperAdmin`
	`vyatta@R1# set service snmp v3 user support group SuperAdmin`

Assign views to an SNMPv3 user group, one view at a time.	`vyatta@R1# set service snmp v3 group EquipmentAuditor view EquipIdent`
	`vyatta@R1# set service snmp v3 group LocalNetAdmin view NetConfig`
	`vyatta@R1# set service snmp v3 group SuperAdmin view TrafficIn`
	`vyatta@R1# set service snmp v3 group SuperAdmin view IntNoDescr`

Commit the change.	`vyatta@R1# commit`

Verify the configuration.	`vyatta@R1:~$ show snmp v3 group`
	
	`SNMPv3 Groups:`
	
	`Group View`
	`----- ----`
	`EquipmentAuditor EquipIdent(ro)`
	`LocalNetAdmin NetConfig(ro)`
	`SuperAdmin IntNoDescr(ro)`

Specifying trap destinations

To configure the destination of trap and inform notifications, specify the IP address (IPv4 or IPv6) of the SNMPv3 trap target and the name of the SNMPv3 user at the trap target. To authenticate the SNMPv3 user at the trap target, specify the **service snmp v3 trap-**

target *addr* **auth** [**encrypted-key** | **plaintext-key**] *passwd* command. To encrypt the notifications at the trap target, specify the **service snmp v3 trap-target** *addr* **privacy** [**encrypted-key** | **plaintext-key**] *passwd* command.

By default, the Brocade vRouter sends inform notifications to trap targets.

The following table shows an example of the Brocade vRouter configured to send SNMP traps to the configured SNMP manager at 172.18.93.86.

To specify trap destinations, SNMPv3 user names at the trap target, and security keys, perform the following steps in configuration mode.

TABLE 19 Specifying trap destinations, SNMPv3 user names and security keys

Step	Command
Specify the IP address of the trap target and the clear-text password used for authenticating the user at the trap target. Note: The clear-text password is stored in the system in this form.	`vyatta@R1# set service snmp v3 trap-target 172.18.93.86 auth plaintext-key password7`
Optional. Specify the IP address of the trap target and the clear-text password used to encrypt the traps and informs. Note: The clear-text password is stored in the system in this form.	`vyatta@R1# set service snmp v3 trap-target 172.18.93.86 privacy plaintext-key password8`
Specify the type of notifications to send to the trap target.	`vyatta@R1#set service snmp v3 trap-target 72.18.93.86 type trap`
Specify the SNMP engine ID of the SNMPv3 trap target. **NOTE** If the **service snmp v3 trap-target** *addr* **type** *type* command is set to trap, you must also the specify the engine ID of the SNMPv3 trap target using **service snmp v3 trap-target** *addr* **engineid** *engineid* command.	`vyatta@R1# set service snmp v3 trap-target 172.18.93.86 engineid 80001f8880634bd405730cdc50`
Specify the IP address of the trap target and the username at the trap target.	`vyatta@R1# set service snmp v3 trap-target 172.18.93.86 user Adminuser`
Commit the change.	`vyatta@R1# commit`

Verify the configuration.	`vyatta@R1:~$ show snmp v3 trap-target`
	`SNMPv3 Trap-targets:`
	`Trap-target Port Protocol Auth`
	`Priv Type EngineID User`
	`---------- ---- -------- ----`
	`---- ---- -------- ----`
	`172.18.93.86 162 udp`
	`md5 trap 80001f8880634bd405730...`
	`Administrator`

SNMP Commands

service snmp

Defines SNMP information for the Brocade vRouter.

Syntax

set service snmp

delete service snmp

show service snmp

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
    }
}
```

Usage Guidelines

Use this command to define information about the SNMP communities to which this system should respond, location of and contact information for the system, and destinations for the SNMP traps.

Use the **set** form of this command to define SNMP settings.

Use the **delete** form of this command to remove all SNMP configuration.

Use the **show** form of this command to view SNMP configuration.

service snmp community <community>

Defines an SNMP community.

Syntax

set service snmp community *community* [**authorization** *auth* | **client** *addr* | **network** *net*]

delete service snmp community *community* [**authorization** | **client** | **network**]

show service snmp community *community* [**authorization** | **client** | **network**]

Command Default

By default, no community is defined.

Parameters

community

Multi-node. An SNMP community. The argument is the community string to be used to authorize SNMP managers making requests of this system. Letters, numbers, and hyphens are supported.
You can define more than one community by creating multiple **community** configuration nodes.

auth

Optional. The privileges for the community. The privileges are as follows:

ro

The community can view system information, but not change it. This is the default privilege.

rw

The community can read and write information.

The default privileges are **ro.**

addr

Optional. Multi-node. The IPv4 or IPv6 address of an SNMP client in the community that is authorized to access the system.
You can define more than one client by creating the **client** configuration node multiple times.

net

Optional. Multi-node. The IPv4 or IPv6 network of SNMP networks in the community that are authorized to access the server.
You can define more than one network by creating the **network** configuration node multiple times.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        community community {
            authorization auth
            client addr
            network net
```

```
            }
        }
}
```

Usage Guidelines

Use this command to define an SNMP community.

If no SNMP clients or networks are explicitly defined, then any client presenting the correct community string is granted the access privilege specified by the authorization option. If a client or network is defined, then only explicitly listed clients or networks are granted access to the system.

Use the **set** form of this command to define an SNMP community.

Use the **delete** form of this command to remove SNMP community configuration or to restore the default value of an option.

Use the **show** form of this command to view SNMP community configuration.

service snmp community <community> view <viewname>

Associates a view with an SNMP community.

Syntax

set service snmp community *community* **view** *viewname*

delete service snmp community *community* **view** *viewname*

show service snmp community *community* **view** *viewname*

Command Default

Not applicable

Parameters

community

The name of an SNMP community.

viewname

The name of a view to be associated with the SNMP community. Only alphanumeric characters for a view name are allowed.

Modes

Configuration mode.

Configuration Statement

```
service {
    snmp {
        community community
        view viewname
    }
}
```

Usage Guidelines

Use this command to associate a view with an SNMP community.

The view must first be defined by using the **service snmp view** *viewname* command.

Use the **set** form of this command to associate a view with an SNMP community.

Use the **delete** form of this command to remove the association between the view and an SNMP community.

Use the **show** form of this command to display the name of the view associated with an SNMP community.

service snmp contact <contact>

Records contact information for the system.

Syntax

set service snmp contact *contact*

delete service snmp contact

show service snmp contact

Parameters

contact

Optional. Contact information for the system. This information is stored as MIB-2 system information. Letters, numbers, and hyphens are supported.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        contact contact
    }
}
```

Usage Guidelines

Use this command to record contact information for the system.

Use the **set** form of this command to record contact information for the system.

Use the **delete** form of this command to remove contact information for the system.

Use the **show** form of this command to view contact information for the system.

service snmp description <desc>

Records a brief description of the system.

Syntax

set service snmp description *desc*

delete service snmp description

show service snmp description

Parameters

desc

Optional. A brief description of the system. This description is stored as MIB-2 system information. Letters, numbers, and hyphens are supported.
When set, this description is stored as the object ID sysDescr. By default, sysDescr is set to Vyatta [*version-string*], where *version-string* is the version of Brocade vRouter software.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        description desc
    }
}
```

Usage Guidelines

Use this command to record a brief description of the system.

Use the **set** form of this command to record a brief description of the system.

Use the **delete** form of this command to remove the system description.

Use the **show** form of this command to view the system description.

service snmp listen-address <addr>

Specifies the IP address on which the SNMP agent listens for requests.

Syntax

set service snmp listen-address *addr* [**port** *port*]

delete service snmp listen-address *addr* [**port**]

show service snmp listen-address *addr* [**port**]

Command Default

The SNMP agent listens on all addresses on port 161.

Parameters

addr

Optional. Multi-node. The IPv4 or IPv6 address on which the SNMP agent listens for requests.
You can specify multiple listening addresses for SNMP by creating multiple **listen-address** configuration nodes.

port

The UDP port used for listening. The default port is 161.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        listen-address addr {
            port port
        }
    }
}
```

Usage Guidelines

Use this command to specify the IPv4 or IPv6 address and port on which the SNMP agent listens for requests.

Use the **set** form of this command to specify listen-address parameters.

Use the **delete** form of this command to remove listen-address parameters.

Use the **show** form of this command to view the listen-address configuration.

service snmp location <location>

Records the location of the system.

Syntax

set service snmp location *location*

delete service snmp location

show service snmp location

Parameters

location

Optional. The location of the system. This location is stored as MIB-2 system information. Letters, numbers, and hyphens are supported.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        location location
    }
}
```

Usage Guidelines

Use this command to record the location of the system.

Use the **set** form of this command to record the location of the system.

Use the **delete** form of this command to remove the system location.

Use the **show** form of this command to view the system location.

service snmp notification

Enables all protocol-specific SNMP traps for the BFD, BGP, or OSPF protocol.

Syntax

set service snmp notification [**bfd** | **bgp** | **ospf**] **all**

delete service snmp notification [**bfd** | **bgp** | **ospf**] **all**

show service snmp

Command Default

All protocol-specific SNMP traps are disabled by default on the system.

Parameters

bfd

Specifies the BFD protocol.

bgp

Specifies the BGP protocol.

ospf

Specifies the OSPF protocol.

Modes

Configuration mode.

Configuration Statement

```
service {
       snmp {
           notification {
             { [ bfd | bgp | ospf ]
                                all
                        }

                  }
        }
}
```

Usage Guidelines

Use this command to enable protocol-specific SNMP traps.

Use the **set** form of this command to enable all protocol-specific SNMP traps for the BFD, BGP, or OSPF protocol.

Use the **delete** form of this command to delete all protocol-specific SNMP traps for the BFD, BGP, or OSPF protocol.

Use the **show** form of this command to display all protocol-specific SNMP traps for the BFD, BGP, or OSPF protocol.

service snmp trap-source <addr>

Specifies the IP address of the source of SNMP traps.

Syntax

set service snmp trap-source *addr*

delete service snmp trap-source *addr*

show service snmp trap-source

Command Default

By default, the system automatically selects the primary IP address of the interface facing the trap target.

Parameters

addr

The IPv4 or IPv6 address of the source of SNMP traps.
This address is included as the source of SNMP traps in SNMP messages sent to an SNMP server. The address must be an address configured on a system interface.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        trap-source addr
    }
}
```

Usage Guidelines

Use this command to specify the IPv4 or IPv6 address of the source of SNMP traps.

Use the **set** form of this command to specify the IP address of the source of SNMP traps.

Use the **delete** form of this command to remove a trap-source address and have the system select the source address automatically.

Use the **show** form of this command to view the trap-source addresses.

service snmp trap-target <addr>

Specifies the IP address and port of the destination for SNMP traps.

Syntax

set service snmp trap-target *addr* [**community** *community* | **port** *port*]

delete service snmp trap-target *addr* [**community** *community* | **port**]

show service snmp trap-target *addr* [**community** *community* | **port**]

Parameters

addr

Optional. Multi-node. The IPv4 or IPv6 address of the destination for SNMP traps.
You can specify multiple destinations for SNMP traps by creating multiple **trap-target** configuration nodes. Or, you can enter a space-separated list of IP addresses.

community

The community used when sending trap information. The default community is **public**.

port

The destination UDP port used for trap notification. The default port is 162.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        trap-target addr {
            community community
            port port
        }
    }
}
```

Usage Guidelines

Use this command to specify the IPv4 or IPv6 address and port of the destination for SNMP traps as well as the community used when sending trap information.

Use the **set** form of this command to specify trap-target parameters.

Use the **delete** form of this command to remove trap-target parameters.

Use the **show** form of this command to view the trap-target configuration.

service snmp view <viewname> oid <oid>

Specifies a subtree to appear in the view.

Syntax

set service snmp view *viewname* **oid** *oid* [**mask** | **exclude**]

delete service snmp view *viewname* **oid** *oid* [**mask** | **exclude**]

show service snmp view *viewname* **oid** *oid*

Command Default

Not applicable

Parameters

viewname

A view.

oid

Multi-node. The Object Identifier (OID) of a subtree to be included in or excluded from the view.

mask

A bit-mask that identifies a single row in a MIB table to be included or excluded. The bitmask is specified as hexadecimal digits delimited with a period (.). For example, ff.a0.

exclude

Exclude the identified subtree.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        view viewname{
            oid oid {
                mask mask
                exclude
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify a subtree to appear in the view.

Use the **set** form of this command to specify a subtree to appear in the view.

Use the **delete** form of this command to remove a specified subtree from the view.

Use the **show** form of this command to view a subtree configuration.

service snmp v3 engineid <engineid>

Specifies the SNMP engine identifier (ID) of an SNMPv3 agent.

Syntax

set service snmp v3 engineid *engineid*

delete service snmp v3 engineid

show service snmp v3 engineid

Parameters

engineid

The engine ID of an SNMP agent. The engine ID consists of 2 to 32 hexadecimal digits.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            engineid engineid
        }
    }
}
```

Usage Guidelines

Use this command to specify the SNMP engine ID of an SNMPv3 agent. This ID is a unique hexadecimal string that is used to identify the SNMP agent for administration purposes. The engine ID is used with a hashing function to generate keys for authentication and encryption of SNMPv3 messages.

CAUTION

If you have SNMPv3 USM users associated with an SNMP engine ID, do not change or delete the value of the SNMP engine ID. The plaintext password that you enter for an SNMPv3 USM user is automatically encrypted using the Message Digest (MD5) encryption. The encrypted password is stored internally for use while the plaintext password is not saved or stored. The encrypted key is based on both the plaintext password and the engine ID. If the engine ID is changed or deleted, the stored encrypted keys for the SNMPv3 users become invalid. You will then be required to add these users to the SNMPv3 configuration once more to have these SNMPv3 users become valid in the Brocade vRouter again.

Use the **set** form of this command to specify the SNMP engine ID of an SNMPv3 agent.

Use the **delete** form of this command to remove the SNMP engine ID of an SNMPv3 agent.

Use the **show** form of this command to view the configuration of the SNMP engine ID of an SNMPv3 agent.

service snmp v3 group <groupname>

Specifies the name of an SNMPv3 user group.

Syntax

set service snmp v3 group *groupname*

delete service snmp v3 group

show service snmp v3 group

Parameters

groupname

The name of an SNMPv3 user group. Only alphanumeric characters are supported.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            group groupname
        }
    }
}
```

Usage Guidelines

Use this command to specify the name of an SNMPv3 user group. Use the **service snmp v3 user** *username* **engineid** *engineid* command to assign a user to a user group.

Use the **set** form of this command to specify the name of an SNMPv3 user group.

Use the **delete** form of this command to remove the name of an SNMPv3 user group.

Use the **show** form of this command to view the name of an SNMPv3 user group.

service snmp v3 group <groupname> mode <mode>

Defines the read/write access for an SNMPv3 user group.

Syntax

set service snmp v3 group *groupname* **mode** *mode*

delete service snmp v3 group *groupname* **mode**

show service snmp v3 group *groupname* **mode**

Command Default

The default mode is **ro**.

Parameters

groupname

The name of an SNMPv3 user group.

mode

The mode for user group access rights. The mode is as follows:

ro

This mode allows users in the user group to view system information, but not change it.

rw

This mode provides users in the user group with read-write privileges.

The default mode is **ro**.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            group groupname {
                mode mode
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the read/write access for an SNMPv3 user group.

Use the **set** form of this command to specify the read/write access for an SNMPv3 user group.

Use the **delete** form of this command to remove the read/write access for an SNMPv3 user group.

Use the **show** form of this command to view the read/write access for an SNMPv3 user group.

service snmp v3 group <groupname> seclevel <seclevel>

Defines the security level to apply to the users within an SNMPv3 user group.

Syntax

set service snmp v3 group *groupname* **seclevel** *seclevel*

delete service snmp v3 group *groupname* **seclevel** *seclevel*

show service snmp v3 group *groupname* **seclevel** *seclevel*

Parameters

groupname

The name of an SNMPv3 user group.

seclevel

The security level for user group. The security level is as follows:

auth

This security level requires users in the user group to use authentication as the security protocol to apply to the SNMP messages sent between an SNMP agent and SNMP manager

priv

This security level requires users in the user group to use encryption as the security protocol to apply to the SNMP messages sent between an SNMP agent and SNMP manager.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            group groupname {
                seclevel seclevel
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the security level to apply to an SNMPv3 user group.

Use the **set** form of this command to specify the security level to apply to an SNMPv3 user group.

Use the **delete** form of this command to remove the security level specified for an SNMPv3 user group.

Use the **show** form of this command to view the security level for an SNMPv3 user group.

service snmp v3 group <groupname> view <viewname>

Associates a view with an SNMPv3 user group.

Syntax

set service snmp v3 group *groupname* **view** *viewname*

delete service snmp v3 group *groupname* **view**

show service snmp v3 group *groupname* **view**

Parameters

groupname

The name of an SNMPv3 user group.

viewname

The name of a view to be associated with the SNMPv3 user group. Only alphanumeric characters for a view name are allowed.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            group groupname {
                view viewname
            }
        }
    }
}
```

Usage Guidelines

Use this command to associate a view with an SNMPv3 user group. The view must first be defined by using the **service snmp v3 view** *viewname* command.

Use the **set** form of this command to associate a view with an SNMPv3 user group.

Use the **delete** form of this command to remove the association between the view and an SNMPv3 user group.

Use the **show** form of this command to display the name of the view associated with an SNMPv3 user group.

service snmp v3 trap-target <addr>

Defines the SNMP target for informs or traps.

Syntax

set service snmp v3 trap-target *addr*

delete service snmp v3 trap-target *addr*

show service snmp v3 trap-target

Parameters

addr

The IPv4 or IPv6 address of the SNMPv3 trap target.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            trap-target addr
        }
    }
}
```

Usage Guidelines

Use this command to define the SNMP target for informs or traps.

Use the **set** form of this command to define the SNMP target for informs or traps.

Use the **delete** form of this command to remove the SNMP target for informs or traps.

Use the **show** form of this command to view the SNMP target for informs or traps.

service snmp v3 trap-target <addr> auth encrypted-key <passwd>

Defines the encrypted password to use for authentication at the trap target.

Syntax

set service snmp v3 trap-target *addr* **auth encrypted-key** *passwd*

delete service snmp v3 trap-target *addr* **auth encrypted-key**

show service snmp v3 trap-target *addr* **auth encrypted-key**

Parameters

addr

The IPv4 or IPv6 address of the SNMPv3 trap target.

passwd

The authentication password. Only hexadecimal passwords are allowed.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            trap-target addr {
                auth {
                    encrypted-key passwd
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to define the encrypted password to use for authentication at the trap target. Use the **service snmp v3 trap-target** *addr* **auth plaintext-key** *passwd* command to specify an unencrypted password for authentication. Only one of these two commands can be used to configure authentication for a given trap target.

Use the **set** form of this command to define the encrypted password for authentication.

Use the **delete** form of this command to remove the encrypted password for authentication.

Use the **show** form of this command to view the encrypted password for authentication.

service snmp v3 trap-target <addr> auth plaintext-key <passwd>

Defines the clear text password used for authentication at the trap target.

Syntax

set service snmp v3 trap-target *addr* **auth plaintext-key** *passwd*

delete service snmp v3 trap-target *addr* **auth plaintext-key**

show service snmp v3 trap-target *addr* **auth plaintext-key**

Parameters

addr

The IPv4 or IPv6 address of the SNMPv3 trap target.

passwd

The authentication password. The password must be eight or more characters. Only alphanumeric characters for a password are allowed.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            trap-target addr {
                auth {
                    plaintext-key passwd
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to define the clear text password used for authentication at the trap target. Use the **service snmp v3 trap-target** *addr* **auth encrypted-key** *passwd* command to specify an encrypted password for authentication. Only one of these two commands can be used to configure authentication for a given trap target.

Use the **set** form of this command to define the clear text password used for authentication.

Use the **delete** form of this command to remove the clear text password for authentication.

Use the **show** form of this command to view the clear text password for authentication.

service snmp v3 trap-target <addr> auth type <type>

Defines the protocol used for authentication at the trap target.

Syntax

set service snmp v3 trap-target *addr* **auth type** *type*

delete service snmp v3 trap-target *addr* **auth type**

show service snmp v3 trap-target *addr* **auth type**

Command Default

The default protocol is **md5.**

Parameters

addr

The IPv4 or IPv6 address of the SNMPv3 trap target.

type

The protocol used for authentication. The protocol is as follows:

md5

Message Digest 5 (MD5) authentication

sha

Secure Hash Algorithm authentication.

The default protocol is **md5.**

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            trap-target addr {
                auth {
                    type type
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to define the protocol used for authentication at the trap target.

Use the **set** form of this command to define the protocol used for authentication.

Use the **delete** form of this command to remove the protocol used for authentication.

Use the **show** form of this command to view the protocol used for authentication.

service snmp v3 trap-target <addr> engineid <engineid>

Specifies the SNMP engine identifier (ID) of the SNMPv3 trap target.

Syntax

set service snmp v3 trap-target *addr* **engineid** *engineid*

delete service snmp v3 trap-target *addr* **engineid**

show service snmp v3 trap-target *addr* **engineid**

Parameters

addr

The IPv4 or IPv6 address of the SNMPv3 trap target.

engineid

The engine ID of the SNMPv3 trap target. The *engineid* consists of 2 to 32 hexadecimal digits.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            trap-target addr {
                engineid engineid
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the SNMP engine ID of the SNMPv3 trap target. This ID is a unique hexadecimal string that is used to identify the SNMP trap target for administration purposes. The engine ID is used with a hashing function to generate keys for authentication and encryption of SNMPv3 messages.

NOTE

If the **service snmp trap-target addr** *addr* command has been set to trap, you must also specify the engine ID of the SNMPv3 trap target using this command.

Use the **set** form of this command to specify the engine ID of the SNMPv3 trap target.

Use the **delete** form of this command to remove the engine ID of the SNMPv3 trap target.

Use the **show** form of this command to view the SNMP engine ID configuration.

service snmp v3 trap-target <addr> port <port>

Specifies the port on a trap target that SNMP traps and to which informs are sent.

Syntax

set service snmp v3 trap-target *addr* **port** *port*

delete service snmp v3 trap-target *addr* **port**

show service snmp v3 trap-target *addr* **port**

Command Default

The trap target uses port 162.

Parameters

addr

The IPv4 or IPv6 address of the SNMPv3 trap target.

port

The port to which SNMPv3 traps and informs are sent. The range of values is 1 to 65535. The default value is 162.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            trap-target addr {
                port port
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the port on a trap target that SNMP traps to which informs are sent.

Use the **set** form of this command to specify the port on a trap target that SNMP traps to which informs are sent.

Use the **delete** form of this command to remove the ports specified and return the system to using the default port.

Use the **show** form of this command to view the port configuration.

service snmp v3 trap-target <addr> privacy encrypted-key <priv-key>

Defines the encrypted key for the privacy protocol used for traps and informs sent to the trap target.

Syntax

set service snmp v3 trap-target *addr* **privacy encrypted-key** *priv-key*

delete service snmp v3 trap-target *addr* **privacy encrypted-key**

show service snmp v3 trap-target *addr* **privacy encrypted-key**

Parameters

addr

The IPv4 or IPv6 address of the SNMPv3 trap target.

priv-key

The privacy key used to encrypt traps and informs sent to the trap target.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            trap-target addr {
                privacy {
                    encrypted-key priv-key
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to define the encrypted key for the privacy protocol used for traps and informs sent to the trap target.

Use the **set** form of this command to define the encrypted key for the privacy protocol used for traps and informs sent to the trap target.

Use the **delete** form of this command to remove the encrypted key for the privacy protocol used for traps and informs sent to the trap target.

Use the **show** form of this command to view the encrypted key for the privacy protocol used for traps and informs sent to the trap target.

service snmp v3 trap-target <addr> privacy plaintext-key <priv-key>

Defines the clear text key for the privacy protocol used for traps and informs sent to the trap target.

Syntax

set service snmp v3 trap-target *addr* **privacy plaintext-key** *priv-key*

delete service snmp v3 trap-target *addr* **privacy plaintext-key** *priv-key*

show service snmp v3 trap-target *addr* **privacy plaintext-key** *priv-key*

Parameters

addr

The IPv4 or IPv6 address of the SNMPv3 trap target.

priv-key

The privacy key used to encrypt traps and informs sent to the trap target. The key must be eight or more characters. Only alphanumeric characters for a privacy key are allowed.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            trap-target addr {
                privacy {
                    plaintext-key priv-key
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to define the clear text key for the privacy protocol used for traps and informs sent to the trap target.

Use the **set** form of this command to define the clear text key for the privacy protocol used for traps and informs sent to the trap target.

Use the **delete** form of this command to remove the clear text key for the privacy protocol used for traps and informs sent to the trap target.

Use the **show** form of this command to view the clear text key for the privacy protocol used for traps and informs sent to the trap target.

service snmp v3 trap-target <addr> privacy type <type>

Defines the protocol used to encrypt traps and informs sent to the trap target.

Syntax

set service snmp v3 trap-target *addr* **privacy type** *type*

delete service snmp v3 trap-target *addr* **privacy type**

show service snmp v3 trap-target *addr* **privacy type**

Command Default

The default value is **des.**

Parameters

addr

The IPv4 or IPv6 address of the SNMPv3 trap target.

type

The protocol used to encrypt traps and informs sent to the trap target. The protocol is as follows:

aes

Advanced Encryption Standard (AES) data encryption.

des

Data Encryption Standard (DES) data encryption

The default value is **des.**

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            trap-target addr {
                privacy {
                    type type
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to define the protocol used to encrypt traps and informs sent to the trap target.

Use the **set** form of this command to define the protocol used for privacy.

Use the **delete** form of this command to remove the protocol used for privacy.

Use the **show** form of this command to view the protocol used for privacy.

service snmp v3 trap-target <addr> protocol <protocol>

Defines the protocol for traps and informs sent to the trap target.

Syntax

set service snmp v3 trap-target *addr* **protocol** *protocol*

delete service snmp v3 trap-target *addr* **protocol**

show service snmp v3 trap-target *addr* **protocol**

Command Default

The system uses UDP.

Parameters

addr

The IPv4 or IPv6 address of the SNMPv3 trap target.

protocol

The protocol used to send traps and informs to the trap target. The protocol is as follows:

tcp

Transmission Control Protocol (TCP).

upd

User Datagram Protocol (UDP).

The default protocol is **udp.**

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            trap-target addr {
                protocol protocol
            }
        }
    }
}
```

Usage Guidelines

Use this command to define the protocol for traps and informs sent to the trap target.

Use the **set** form of this command to define the protocol for traps and informs sent to the trap target.

Use the **delete** form of this command to remove the protocol for traps and informs sent to the trap target.

Use the **show** form of this command to view the protocol for traps and informs sent to the trap target.

service snmp v3 trap-target <addr> type <type>

Specifies the type of notifications to send to the trap target.

Syntax

set service snmp v3 trap-target *addr* **type** *type*

delete service snmp v3 trap-target *addr* **type**

show service snmp v3 trap-target *addr* **type**

Command Default

The system uses **inform**.

Parameters

addr

The IPv4 or IPv6 address of the SNMPv3 trap target.

type

The notification type. The notification type is as follows:

inform

An SNMPv3 message sent to the trap target that requires acknowledgment.

trap

An SNMPv3 message sent to the trap target that does not require acknowledgment.
The default notification type is **inform.**

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            trap-target addr {
                type type
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the type of notifications to send to the trap target.

NOTE

If the notification type of trap is set using this command, then you must also specify the engine ID of the SNMPv3 trap target using the **service snmp v3 trap-target** *addr* **engineid** *engineid* command.

Use the **set** form of this command to specify the type of notifications sent to the trap target.

Use the **delete** form of this command to return the system to its default notification type.

Use the **show** form of this command to view the type of notifications sent to the trap target.

service snmp v3 trap-target <addr> user <username>

Defines an SNMPv3 username for authentication at the trap target.

Syntax

set service snmp v3 trap-target *addr* **user** *username*

delete service snmp v3 trap-target *addr* **user**

show service snmp v3 trap-target *addr* **user**

Parameters

addr

The IPv4 or IPv6 address of the SNMPv3 trap target.

username

The name of an SNMPv3 user at the trap target.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            trap-target addr {
                user username
            }
        }
    }
}
```

Usage Guidelines

Use this command to define a username for authentication at the trap target.

Use the **set** form of this command to define a username for authentication at the trap target.

Use the **delete** form of this command to remove a username for authentication.

Use the **show** form of this command to view a username for authentication.

service snmp v3 tsm

Specifies that SNMPv3 uses the Transport Security Model (TSM).

Syntax

set service snmp v3 tsm

delete service snmp v3 tsm

show service snmp v3 tsm

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            tsm
        }
    }
}
```

Usage Guidelines

Use this command to specify that SNMPv3 uses TSM for encryption.

Use the **set** form of this command to specify that SNMPv3 uses TSM encryption.

Use the **delete** form of this command to remove TSM encryption for SNMPv3.

Use the **show** form of this command to view that SNMPv3 uses TSM encryption.

service snmp v3 tsm local-key <local-key>

Specifies the fingerprint of a Transport Security Model (TSM) certificate for a server.

Syntax

set service snmp v3 tsm local-key *local-key*

delete service snmp v3 tsm local-key

show service snmp v3 tsm local-key

Parameters

local-key

The fingerprint of a TSM certificate or the filename of a key file.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            tsm {
                local-key local-key
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the fingerprint of a TSM certificate for a server. The fingerprint can be specified either directly through a TSM certificate or indirectly through the name of the file containing the fingerprint.

Use the **set** form of this command to specify the fingerprint of a TSM certificate for a server.

Use the **delete** form of this command to remove the fingerprint of a TSM certificate for a server.

Use the **show** form of this command to view the fingerprint of a TSM certificate for a server.

service snmp v3 tsm port <port>

Defines the port used for TSM.

Syntax

set service snmp v3 tsm port *port*

delete service snmp v3 tsm port

show service snmp v3 tsm port

Command Default

The system uses port 10161.

Parameters

port

The port used for TSM. The range of values is 1 to 65535. The default value is 10161.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            tsm {
                port port
            }
        }
    }
}
```

Usage Guidelines

Use this command to define the port used for TSM.

Use the **set** form of this command to define the port used for TSM.

Use the **delete** form of this command to remove the port for TSM.

Use the **show** form of this command to view the port for TSM.

service snmp v3 user <username> auth encrypted-key <passwd>

Defines the encrypted password used to authenticate a user.

Syntax

set service snmp v3 user *username* **auth encrypted-key passwd**

delete service snmp v3 user *username* **auth encrypted-key**

show service snmp v3 user *username* **auth encrypted-key**

Parameters

username

The name of an SNMPv3 user.

passwd

The authentication password. Only hexadecimal passwords are allowed.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            user username {
                auth {
                    encrypted-key passwd
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to define the encrypted password used to authenticate a user.

Use the **set** form of this command to define the encrypted password used to authenticate a user.

Use the **delete** form of this command to remove the encrypted password used to authenticate a user.

Use the **show** form of this command to view the encrypted password used to authenticate a user.

service snmp v3 user <username> auth plaintext-key <passwd>

Defines the clear text password used to authenticate a user.

Syntax

set service snmp v3 user *username* **auth plaintext-key** *passwd*

Parameters

username

The name of an SNMPv3 user.

passwd

The authentication password. The password must be eight or more characters. Only alphanumeric characters for a password are allowed.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            user username {
                auth {
                    plaintext-key passwd
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to define the clear text password used to authenticate a user.

NOTE

The plaintext password that you enter for an SNMPv3 user is automatically encrypted using the Message Digest (MD5) encryption. The encrypted password is stored internally for use while the plaintext password is not saved or stored.

Use the **set** form of this command to define the clear text password used to authenticate a user.

service snmp v3 user <username> auth type <type>

Defines the protocol used for user authentication.

Syntax

set service snmp v3 user *username* **auth type** *type*

delete service snmp v3 user *username* **auth type**

show service snmp v3 user *username* **auth type**

Command Default

The system uses MD5.

Parameters

username

The name of an SNMPv3 user.

type

The protocol used for user authentication. The protocol is as follows:

md5

Message Digest 5 (MD5) authentication.

sha

Secure Hash Algorithm (SHA) authentication.

The default protocol is **md5**.

Modes

Configuration mode

Usage Guidelines

Use this command to define the protocol used for user authentication.

Use the **set** form of this command to define the protocol used for user authentication.

Use the **delete** form of this command to remove the protocol used for user authentication.

Use the **show** form of this command to view the protocol used for user authentication.

Configuration Statement

```
service {
    snmp {
        v3 {
            user username {
                auth {
                    type type
                }
            }
        }
```

```
        }
}
```

service snmp v3 user <username> engineid <engineid>

Specifies the SNMP engine ID of an SNMPv3 user.

Syntax

set service snmp v3 user *username* **engineid** *engineid*

delete service snmp v3 user *username* **engineid** *engineid*

show service snmp v3 user *username* **engineid** *engineid*

Parameters

username

The name of an SNMPv3 user.

engineid

The engine ID of an SNMPv3 user. The engine ID consists of 2 to 32 hexadecimal digits.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            user username {
                engineid engineid
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the SNMPv3 engine ID of an SNMPv3 user. This ID is a unique hexadecimal string that is used to identify the SNMPv3 user for administration purposes. The engine ID is used with a hashing function to generate keys for authentication and encryption of SNMP v3 messages.

CAUTION

If you have SNMPv3 USM users associated with an SNMPv3 engine ID within your SNMPv3 configuration, do not change or delete the value of the SNMPv3 engine ID. The plaintext password that you enter for an SNMPv3 USM user is automatically encrypted using the Message Digest (MD5) encryption. The encrypted password is stored internally for use while the plaintext password is not saved or stored. The encrypted key is based on both the plaintext password and engine ID. If the engine ID is changed or deleted, the stored encrypted keys for the SNMPv3 USM users become invalid. You will then be required to add these users to the SNMPv3 configuration once more to have these SNMPv3 users become valid in the Brocade vRouter again.

Use the **set** form of this command to specify the engine ID of an SNMPv3 user.

Use the **delete** form of this command to remove the engine ID of an SNMPv3 user.

Use the **show** form of this command to view the SNMPv3 engine ID configuration of SNMPv3 users.

service snmp v3 user <username> group <groupname>

Assigns an SNMPv3 user to a user group.

Syntax

set service snmp v3 user *username* **group** *groupname*

delete service snmp v3 user *username* **group**

show service snmp v3 user *username* **group**

Parameters

username

The name of an SNMPv3 user.

groupname

The name of a user group.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            user username {
                group groupname
            }
        }
    }
}
```

Usage Guidelines

Use this command to assign an SNMPv3 user to a user group. The user group must first be created by using the **service snmp v3 group** *groupname* command.

Use the **set** form of this command to assign an SNMPv3 user to a user group.

Use the **delete** form of this command to remove an SNMPv3 user from a user group.

Use the **show** form of this command to view the user group to which an SNMPv3 user is assigned.

service snmp v3 user <username> mode <mode>

Specifies the mode for user access rights.

Syntax

set service snmp v3 user *username* **mode** *mode*

delete service snmp v3 user *username* **mode**

show service snmp v3 user *username* **mode**

Command Default

The default mode is **ro**.

Parameters

username

The name of an SNMPv3 user.

mode

The mode for user access rights. The mode is as follows:

ro

This mode allows a user to view system information, but not change it.

rw

This mode provides a user with read-write privileges.

The default mode is **ro.**

Modes

Configuration mode

Usage Guidelines

Use this command to specify the mode for user access rights.

Use the **set** form of this command to specify the mode for user access rights.

Use the **delete** form of this command to remove the mode for user access rights.

Use the **show** form of this command to view the mode for user access rights.

Configuration Statement

```
service {
    snmp {
        v3 {
            user username {
                mode mode
            }
        }
    }
}
```

service snmp v3 user <username> privacy encrypted-key <priv-key>

Defines the encrypted key used to encrypt user traffic.

Syntax

set service snmp v3 user *username* **privacy encrypted-key** *priv-key*

delete service snmp v3 user *username* **privacy encrypted-key**

show service snmp v3 user *username* **privacy encrypted-key**

Parameters

username

The name of an SNMPv3 user.

priv-key

The encrypted key. Only hexadecimal keys are supported.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            user username {
                privacy {
                    encrypted-key priv-key
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the encrypted key used to encrypt user traffic.

Use the **set** form of this command to specify the encrypted key used to encrypt user traffic.

Use the **delete** form of this command to remove the encrypted key.

Use the **show** form of this command to view the encrypted key used to encrypt user traffic.

service snmp v3 user <username> privacy plaintext-key <priv-key>

Defines the clear text key used to encrypt user traffic.

Syntax

set service snmp v3 user *username* **privacy plaintext-key** *priv-key*

Parameters

username

The name of an SNMPv3 user.

priv-key

The clear text key.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            user username {
                privacy {
                    plaintext-key priv-key
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to define the clear text key used to encrypt user traffic.

NOTE

The plaintext password that you enter for an SNMPv3 user is automatically encrypted using the Message Digest (MD5) encryption. The encrypted password is stored internally for use while the plaintext password is not saved or stored.

Use the **set** form of this command to define the clear text key used to encrypt user traffic.

service snmp v3 user <username> privacy type <type>

Defines the protocol used to encrypt user traffic.

Syntax

set service snmp v3 user *username* **privacy type** *type*

delete service snmp v3 user *username* **privacy type**

show service snmp v3 user *username* **privacy type**

Command Default

The system uses **aes**.

Parameters

username

The name of an SNMPv3 user.

type

The protocol used to encrypt user traffic. The protocol is as follows:

aes

Advanced Encryption Standard (AES) data encryption.

des

Data Encryption Standard (DES) data encryption.

The default protocol is **aes.**

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            user username {
                privacy {
                    type type
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to define the protocol used to encrypt user traffic.

Use the **set** form of this command to define the protocol used to encrypt user traffic.

Use the **delete** form of this command to remove the protocol used to encrypt user traffic.

Use the **show** form of this command to view the protocol used to encrypt user traffic.

service snmp v3 user <username> tsm-key <key>

Specifies the fingerprint of or the file containing the Transport Security Model (TSM) certificate.

Syntax

set service snmp v3 user *username* **tsm-key** *key*

delete service snmp v3 user *username* **tsm-key**

show service snmp v3 user *username* **tsm-key**

Parameters

username

The name of an SNMPv3 user.

key

The fingerprint of or the file containing the TSM certificate.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            user username {
                tsm-key key
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the fingerprint of or the file containing the TSM certificate for a user.

Use the **set** form of this command to specify the fingerprint of or the file containing the TSM certificate.

Use the **delete** form of this command to remove the fingerprint or file name of the TSM certificate.

Use the **show** form of this command to view the fingerprint or file name of the TSM certificate.

service snmp v3 view <viewname>

Creates a view.

Syntax

set service snmp v3 view *viewname*

delete service snmp v3 view *viewname*

show service snmp v3 view

Parameters

viewname

The name of a view.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            view viewname
        }
    }
}
```

Usage Guidelines

Use this command to create a view. A view specifies which objects a user can see.

Use the **set** form of this command to create a view.

Use the **delete** form of this command to remove a view.

Use the **show** form of this command to show the view.

service snmp v3 view <viewname> oid <oid>

Specifies a subtree to appear in the view.

Syntax

set service snmp v3 view *viewname* **oid** oid [**mask** mask | **exclude**]

delete service snmp v3 view *viewname* **oid** *oid* [**mask** | **exclude**]

show service snmp v3 view view *viewname* **oid** *oid*

Parameters

viewname

The name of a view.

oid

Multi-node. The Object Identifier (OID) of a subtree to be included in or excluded from the view.

mask

A bit-mask that identifies a single row in a MIB table to be included or excluded. The bitmask is specified as hexadecimal digits delimited with a period (.). For example, ff.a0.

exclude

Exclude the identified subtree.

Modes

Configuration mode

Configuration Statement

```
service {
    snmp {
        v3 {
            view viewname {
                oid oid {
                    mask mask
                    exclude
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify a subtree to appear in the view.

Use the **set** form of this command to specify a subtree to appear in the view.

Use the **delete** form of this command to remove a specified subtree from the view.

Use the **show** form of this command to view a subtree configuration.

show snmp

Displays SNMP statistics.

Syntax

show snmp [**community** | **mib** | **v3**]

Modes

Operational mode

Parameters

community

Status of SNMP community.

mib

SNMP MIB information.

v3

Status of SNMP v3 on local host.

Usage Guidelines

Use this command to display SNMP statistics.

Examples

The following example shows the output for **show snmp**.

```
vyatta@R1:~$ show snmp
[UDP: [127.0.0.1]:161->[0.0.0.0]]=>[Vyatta 999.larkspurse.06200031] Up: 0:02:40.80
Interfaces: 5, Recv/Trans packets: 545097/179020 | IP: 202587/89811
vyatta@R1:~$
```

show snmp v3 certificates

Displays TSM certificates.

Syntax

show snmp v3 certificates

Modes

Operational mode.

Usage Guidelines

Use this command to display TSM certificates.

Examples

The following example shows the output for **show snmp v3 certificates**.

```
vyatta@R1:~$ show snmp v3 certificates
/etc/snmp/tls:

certs/snmpd.crt:
subject=
/C=US/ST=CA/L=Davis/O=Net-SNMP/OU=Development/CN=vyatta@debian/emailAddress=vyatta@debian
SHA1 Fingerprint=33:F2:92:24:E8:1A:D4:99:10:91:F5:A8:84:2A:2E:AD:96:C7:FE:C0

certs/usertsmro.crt:
subject=
/C=US/ST=CA/L=Davis/O=Net-SNMP/OU=Development/CN=vyatta@debian/emailAddress=vyatta@debian
SHA1 Fingerprint=15:6B:82:AB:FA:27:1F:E0:1C:1D:5B:F4:0E:2E:41:A0:C6:38:3E:11

certs/usertsmrw.crt:
subject=
/C=US/ST=CA/L=Davis/O=Net-SNMP/OU=Development/CN=vyatta@debian/emailAddress=vyatta@debian
SHA1 Fingerprint=CE:4A:F4:48:D7:44:B6:9E:F5:1D:05:F9:66:C7:C0:DE:9D:98:08:9E

vyatta@R1:~$
```

show snmp v3 group

Displays a list of configured groups.

Syntax

show snmp v3 group

Modes

Operational mode

Usage Guidelines

Use this command to display a list of configured groups.

Examples

The following example shows the output for **show snmp v3 group**.

```
vyatta@R1:~$ show snmp v3 group
SNMPv3 Groups:
Group               View
-----               ----
group1              view1(ro)
group2              view2(ro)
group3              view3(ro)
vyatta@R1:~$
```

show snmp v3 trap-target

Displays a list of configured targets.

Syntax

show snmp v3 trap-target

Modes

Operational mode

Usage Guidelines

Use this command to display a list of configured targets.

Examples

The following example shows the output for **show snmp v3 trap-target**.

```
vyatta@R1:~$ show snmp v3 trap-target
SNMPv3 Trap-targets:
Trap-target                  Port   Protocol Auth Priv Type   EngineID       User
-----------                  ----   -------- ---- ---- ----   --------       ----
1.1.1.1                      12345  udp      md5       inform                user1111
1.1.1.2                      162    udp      sha  des  inform                user1112
1.1.1.3                      162    udp      md5       trap   0123456abcdf   user1113
vyatta@R1:~$
```

show snmp v3 user

Displays a list of configured users.

Syntax

show snmp v3 user

Modes

Operational mode

Usage Guidelines

Use this command to display a list of configured users.

Examples

The following example shows the output for **show snmp v3 user**.

```
vyatta@R1:~$ show snmp v3 user
SNMPv3 Users:
User                Auth Priv Mode Group
----                ---- ---- ---- -----
user1               md5       ro   group1
user2               md5  aes  ro   group2
user3               sha       ro   group3
user4               md5       rw
vyatta@R1:~$
```

show snmp v3 view

Displays a list of configured views.

Syntax

show snmp v3 view

Modes

Operational mode

Usage Guidelines

Use this command to display a list of configured views.

Examples

The following example shows the output for **show snmp v3 view**.

```
vyatta@R1:~$ show snmp v3 view
SNMPv3 Views:
View : view1
OIDs :
       .1.1.1.1.1.1
       .1.1.1.1.1.2
       .1.1.1.1.1.3 mask ff.a0
View : view2
OIDs :
       .2.1.1.1.1
       .2.1.1.1.1.2 exclude
View : view3
OIDs :
       .3.1.1.1.1
       .3.1.1.1.1.1 exclude
vyatta@R1:~$
```

List of Acronyms

Acronym	Description
ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
AH	Authentication Header
AMI	Amazon Machine Image
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
AWS	Amazon Web Services
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CCMP	AES in counter mode with CBC-MAC
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol version 6
DLCI	data-link connection identifier
DMI	desktop management interface
DMVPN	dynamic multipoint VPN
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
EBS	Amazon Elastic Block Storage
EC2	Amazon Elastic Compute Cloud
EGP	Exterior Gateway Protocol
ECMP	equal-cost multipath
ESP	Encapsulating Security Payload
FIB	Forwarding Information Base
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
I/O	Input/Output
ICMP	Internet Control Message Protocol

Acronym	Description
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IGMP	Internet Group Management Protocol
IGP	Interior Gateway Protocol
IPS	Intrusion Protection System
IKE	Internet Key Exchange
IP	Internet Protocol
IPOA	IP over ATM
IPsec	IP Security
IPv4	IP Version 4
IPv6	IP Version 6
ISAKMP	Internet Security Association and Key Management Protocol
ISM	Internet Standard Multicast
ISP	Internet Service Provider
KVM	Kernel-Based Virtual Machine
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
LDAP	Lightweight Directory Access Protocol
LLDP	Link Layer Discovery Protocol
MAC	medium access control
mGRE	multipoint GRE
MIB	Management Information Base
MLD	Multicast Listener Discovery
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
NBMA	Non-Broadcast Multi-Access
ND	Neighbor Discovery
NHRP	Next Hop Resolution Protocol
NIC	network interface card
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OSPFv2	OSPF Version 2
OSPFv3	OSPF Version 3
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PAT	Port Address Translation
PCI	peripheral component interconnect
PIM	Protocol Independent Multicast
PIM-DM	PIM Dense Mode

Acronym	Description
PIM-SM	PIM Sparse Mode
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PTMU	Path Maximum Transfer Unit
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RHEL	Red Hat Enterprise Linux
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
RP	Rendezvous Point
RPF	Reverse Path Forwarding
RSA	Rivest, Shamir, and Adleman
Rx	receive
S3	Amazon Simple Storage Service
SLAAC	Stateless Address Auto-Configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SPT	Shortest Path Tree
SSH	Secure Shell
SSID	Service Set Identifier
SSM	Source-Specific Multicast
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TBF	Token Bucket Filter
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service
TSS	TCP Maximum Segment Size
Tx	transmit
UDP	User Datagram Protocol
VHD	virtual hard disk
vif	virtual interface
VLAN	virtual LAN
VPC	Amazon virtual private cloud
VPN	virtual private network
VRRP	Virtual Router Redundancy Protocol

Acronym	Description
WAN	wide area network
WAP	wireless access point
WPA	Wired Protected Access