VYATTA, INC. | Vyatta System

Basic System

REFERENCE GUIDE

Using the CLI
Working with Configuration
System Management
User Management
Flow Accounting
Connection Tracking
Logging
SNMP

Vyatta
Suite 200
1301 Shoreway Road
Belmont, CA 94002
vyatta.com
650 413 7200
1 888 VYATTA 1 (US and Canada)

COPYRIGHT

Vyatta reserves the right to make changes to software, hardware, and documentation without notice. For the most recent version of documentation, visit the Vyatta web site at vyatta.com.

PROPRIETARY NOTICES

RELEASE DATE: July 2011

DOCUMENT REVISION. R6.3 v01

RELEASED WITH: R6.3.0

PART NO. A0-0210-10-0012

Contents

Quick List of Examples

Use this list to help you locate examples you'd like to look at or try.

Quick List of Commands

Use this list to help you quickly locate commands.

Preface

This document describes the various deployment, installation, and upgrade options for Vyatta software.

This preface provides information about using this guide. The following topics are presented:

- Intended Audience
- Organization of This Guide
- Document Conventions
- Vyatta Publications

Intended Audience

This guide is intended for experienced system and network administrators. Depending on the functionality to be used, readers should have specific knowledge in the following areas:

- Networking and data communications
- TCP/IP protocols
- General router configuration
- Routing protocols
- Network administration
- Network security
- IP services

Organization of This Guide

This guide has the following aid to help you find the information you are looking for:

- Quick List of Commands

  Use this list to help you quickly locate commands.

- Quick List of Examples

  Use this list to help you locate examples you'd like to try or look at.

This guide has the following chapters:

Chapter	Description	Page
Chapter 1: Using the CLI	This chapter provides an overview of the Vyatta command-line interface (CLI), which is the primary user interface to the Vyatta system, and the operational mode of the CLI.	1
Chapter 2: Working with Configuration	This chapter describes configuration management utilities on the Vyatta system.	25
Chapter 3: System Management	This chapter describes Vyatta system features for basic system management tasks, such as setting host information, working with the ARP cache, and setting the system date and time.	98
Chapter 4: User Management	This chapter explains how to set up user accounts and user authentication.	187

Document Conventions

This guide uses the following advisory paragraphs, as follows.

WARNING *Warnings alert you to situations that may pose a threat to personal safety.*

CAUTION *Cautions alert you to situations that might cause harm to your system or damage to equipment, or that may affect service.*

NOTE *Notes provide information you might need to avoid problems or configuration errors.*

This document uses the following typographic conventions.

`Monospace`	Examples, command-line output, and representations of configuration nodes.
`bold Monospace`	Your input: something you type at a command line.
bold	Commands, keywords, and file names, when mentioned inline. Objects in the user interface, such as tabs, buttons, screens, and panes.
italics	An argument or variable where you supply a value.
<key>	A key on your keyboard, such as <Enter>. Combinations of keys are joined by plus signs ("+"), as in <Ctrl>+c.
[key1 | key2]	Enumerated options for completing a syntax. An example is [enable | disable].
num1–numN	A inclusive range of numbers. An example is 1–65535, which means 1 through 65535, inclusive.
arg1..argN	A range of enumerated values. An example is eth0..eth3, which means eth0, eth1, eth2, or eth3.
arg[*arg*...] *arg*[,*arg*...]	A value that can optionally represent a list of elements (a space-separated list and a comma-separated list, respectively).

Vyatta Publications

Full product documentation is provided in the Vyatta technical library. To see what documentation is available for your release, see the *Guide to Vyatta Documentation*. This guide is posted with every release of Vyatta software and provides a great starting point for finding the information you need.

Additional information is available on www.vyatta.com and www.vyatta.org.

Chapter 1: Using the CLI

This chapter provides an overview of the Vyatta command-line interface (CLI), which is the primary user interface to the Vyatta system, and the operational mode of the CLI.

Note: Configuration using the CLI is discussed in Chapter 2: Working with Configuration.

This chapter presents the following topics:

- CLI Features
- Operational Commands
- Basic Commands for Using the CLI

CLI Features

This section presents the following topics:

- Command Modes
- The Vyatta CLI and the System Shell
- Accessing the CLI
- The Predefined User Account
- User Privilege Levels
- Command Prompts
- Using Special Characters in Commands
- Command Completion
- Command History
- Command Editing
- Filtering Command Output
- Operational Commands
- Running an Operational Command in Configuration Mode

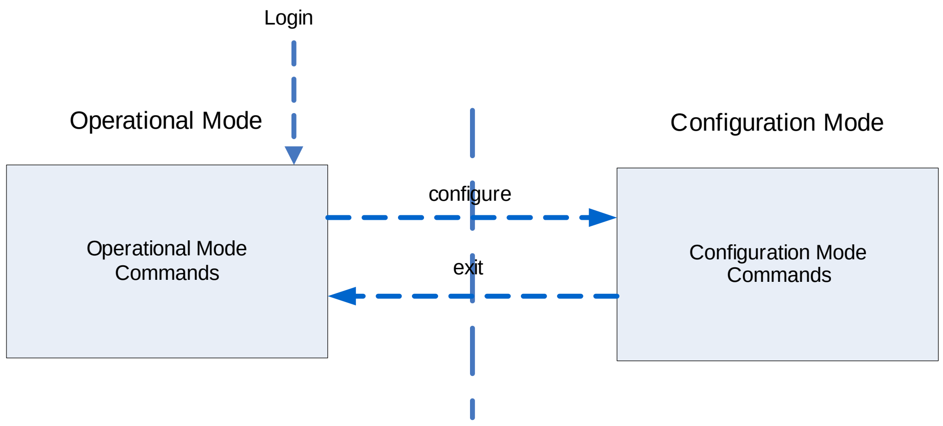
Command Modes

There are two command modes in the Vyatta CLI: operational mode and configuration mode.

Operational mode provides access to operational commands for showing and clearing information and enabling or disabling debugging, as well as commands for configuring terminal settings, loading and saving configuration, and restarting the system. When you log on to the system, the system is in operational mode.

Figure 1-1 shows Vyatta CLI command modes.

Figure 1-1 CLI Command Modes



Configuration mode provides access to commands for creating, modifying, deleting, committing and showing configuration information, as well as commands for navigating through the configuration hierarchy.

- To enter configuration mode from operational mode, issue the **configure** command.
- To return to operational mode from configuration mode, issue the **exit** command. If there are uncommitted configuration changes, you must either commit the changes using the **commit** command, or discard the changes using the **discard** command (or **exit discard**), before you can exit to operational mode. If you have not saved the configuration (using the **save** command) you will be warned that configuration changes have not been saved. When the system is restarted it will load the last saved configuration.

Issuing the **exit** command in operational mode logs you out of the system.

The Vyatta CLI and the System Shell

The CLI of the Vyatta system includes two kinds of commands:

- Vyatta-specific commands for operating and configuring the Vyatta system.
- Commands provided by the operating system shell in which the Vyatta CLI operates.

The commands you can execute depend on your user role and its privileges. However, any command you have the privileges to execute, including operating system commands, can be executed from within the Vyatta CLI.

Accessing the CLI

To access the command-line interface (CLI), you log on to the Vyatta system, either directly through the VGA console, a serial console, or remotely using an SSH or Telnet session. The VGA console also provides nine virtual console sessions. These virtual consoles (tty1-tty9) can be accessed using the key combinations **ALT-F1** (for tty1) through **ALT-F9** (for tty9). tty1 through tty6 will provide a login prompt. tty7 through tty9 are not used.

Regardless of the access method you choose, after the startup messages complete, the login prompt appears, as follows:

```
vyatta login:
```

Log on using the ID and password of a defined user account.

NOTE *You can change user accounts using operating system commands, but the changes will not persist across reboots. For persistent changes to user account information, use the Vyatta CLI.*

The Predefined User Account

By default, the system has one predefined user account, the user **vyatta**. The default password for the vyatta account is **vyatta**. The **vyatta** user has administrator-level privileges and can execute all Vyatta commands and all operating system commands. Note that, although the user can execute both Vyatta commands and operating system commands, command completion and CLI help show only Vyatta commands (for clarity).

User Privilege Levels

The Vyatta system supports two user roles:

- "Admin" Users
- "Operator"Users

"Admin" Users

Admin users have full access to the Vyatta CLI. Admin users can view, configure, and delete information, and execute all Vyatta operational commands. Admin users can also execute all operating system shell commands and constructs.

The default user **vyatta** is an admin user.

To create an admin user, issue the following set of commands in configuration mode:

```
vyatta@vyatta# set system login user user-name level admin
vyatta@vyatta# set system login user user-name authentication
plaintext-password password
```

```
vyatta@vyatta# commit
```

where *user-name* is the ID of the user account you want to create and *password* is the password you are assigning to the user.

Although operating system shell commands are always available to admin users, they are not shown when these users use command completion to query the CLI for available commands. This is because there are several hundred operating system shell commands and constructs available at any time: showing all available operating system shell commands would make it very difficult to distinguish available Vyatta CLI commands.

Admin users can see available commands by entering **help** at the command prompt.

You can remove the restriction on command completion by setting the **VYATTA_RESTRICTED_MODE** environment variable to **none**:

```
export VYATTA_RESTRICTED_MODE=none
```

This removes command completion restriction for all users, regardless of privilege level.

“Operator”Users

Operator users have read-only access to configuration plus the ability to execute Vyatta operational commands. Operator users can view in operational mode (using **show** commands), configure their terminal settings (using the **terminal** command), and exit from the Vyatta CLI (using the **exit** command). Operator users cannot enter configuration mode; however they can display configuration by issuing the **show configuration** command in operational mode.

Basic commands for displaying information (for example, **show configuration** plus the “**pipe**” commands, such as **more**, for managing display output) are available. Commands that use control constructs (such as **if**, **for**, and so on), list operators (such as “;”, “&&”, and so on), and redirection are not available to operator users.

To create an operator user, issue the following command:

```
vyatta@vyatta# set system login user user-name level operator
vyatta@vyatta# set system login user user-name authentication
plaintext-password password
vyatta@vyatta# commit
```

where *user-name* is the ID of the user account you are creating and *password* is the password you are assigning to the user.

Operating system shell commands are not available to operator users and consequently, the list of commands returned using command completion for operator-level users is restricted to Vyatta commands.

You can remove the restriction on command completion by setting the **VYATTA_RESTRICTED_MODE** environment variable to **none**, as follows:

```
export VYATTA_RESTRICTED_MODE=none
```

This removes command completion restriction for all users, regardless of privilege level.

Command Prompts

The command prompt shows you what user account you are logged on under, the hostname of the system you are logged onto, and whether you are in configuration mode or operational mode.

The format of the command prompt in configuration mode as follows:

```
username@hostname#
```

The format of the command prompt in operational mode is as follows:

```
username@hostname:~$
```

where, in both cases, *username* is the user account under which you are logged on and *hostname* is the hostname configured for the system; see Table 1-1 for examples.

Table 1-1 Command prompts

The prompt shows this	And means this
vyatta@R1:~$	User: vyatta Hostname: R1 Command mode: Operational mode
vyatta@R1#	User: vyatta Hostname: R1 Command mode: Configuration mode

Using Special Characters in Commands

The Vyatta FusionCLI management interface is based on the GNU Bash shell. When entering a command at the command prompt, keep in mind that some characters have special meaning to the shell. For example, one such special character is the space character, which denotes the end of a token in a command, as shown below

```
prompt> show interfaces ethernet
```

In this example, the space characters separate the command line into three components: “show,” “interfaces,” and “ethernet.”

If you want to enter string that includes a literal character understood by the shell as a special character, you must enclose the character in double quotation marks. For example, if you want to enter a string that includes a space, you must enclose the string in double quotation marks as shown below:

```
vyatta@vyatta# set firewall name TEST description "external inbound"
```

In this example, the space within the string “external inbound” is within quotes and therefore loses its special meaning as a token separator.

Another example of a special character is the “pipe” character (also called the vertical bar, “|”), which separates two commands and means that the output of the left-hand side command should be processed using the right-hand side command, as shown in the following example:

```
vyatta@vyatta# show interfaces | match eth
```

In this example, the pipe character tells the shell to execute the **show interfaces** command and then process the output using the **match eth** command; as a result, only lines that contain the string “eth” will be displayed. As for the space character, if you want a literal vertical bar in a command component, you must enclose it in double quotation marks.

In addition to the space and vertical bar, the following characters have special meaning for the shell:

- ampersand (“&”)
- semi-colon (“;”)
- comma (“,”)
- left parenthesis (“(“)
- right parenthesis (“)”)
- left angle bracket (“<”)
- right angle bracket (“>”)
- backslash (“\”)
- pound sign (“#”)

In general, if you are unsure what characters are special, a good rule of thumb is to enclose anything that is not alphanumeric within double quotation marks.

Note that within a quoted string, you can include a literal quote mark by preceding it with a backslash, as follows:

```
"some \"quotes\" within quotes"
```

Of course, the rules become more complex if you want a literal backslash. As a general rule, try to avoid using quotataion marks or backslashes as literal configuration values.

Command Completion

You can have the system auto-complete a command syntax by entering any of the following at the command prompt:

Table 1-2 CLI Help Keystrokes

Type this:	To see this:
<Tab>	Auto-completes a command. • If the command is unambiguous, the system generates the next token in the syntax. • If more than one completion is possible, the system displays the set of next possible tokens. (Note that the space following a command or keyword counts as a token.) Pressing <Tab> a second time generates CLI help for the current set of tokens.
?	Pressing the question mark key (“?”) also generates command completion. To enter a literal question mark, first enter <Ctrl>+v, then the question mark.
<Tab> <Alt>-?	Displays all available Vyatta commands and provides command completion.

The following example finds all available commands.

```
vyatta@R1:~$ <Tab>
```

The following example requests command completion for the typed string **sh**. In this example, the command to be completed is unambiguous.

```
vyatta@R1~$ sh<Tab>
```

```
vyatta@R1~$ show
```

The following example requests command completion for the typed string **s**. In this case, there is more than one command that could complete the entry and the system lists all valid completions.

```
vyatta@R1~$:s<Tab>
set       show
```

Note that neither the <Tab> key nor the <Alt>+? key combination provides a help function when double-quoted. When used within double quotes, the <Tab> key generates a tab character and the <Alt>+? key combination generates a question mark (“?”) character.

Command History

The Vyatta System shell supports a command history, where commands you execute are stored in an internal buffer and can be re-executed or edited.

Table 1-3 shows the most important history keystrokes.

Table 1-3 Command history keystrokes

Type this:	To do this:
<Up-Arrow> <Control>-p	Move to the previous command.
<Down-Arrow> <Control>-n	Move to the next command.

Command Editing

The Vyatta System shell supports emacs-style command editing.

Table 1-4 shows the most important editing keystrokes.:

Table 1-4 Command-line editing keystrokes

Type this:	To do this:
<Left-Arrow> <Control>-b	Move backward in the command line.

Table 1-4 Command-line editing keystrokes

Type this:	To do this:
<Right-Arrow> <Control>-f	Move forward in the command line.
<Control>-a	Move to the beginning of the command line.
<Control>-e	Move the end of the command line.
<Control>-d	Delete the character directly under the cursor.
<Control>-t	Toggle (swap) the character under the cursor with the character immediately preceding it.
<Control>-<Space>	Mark the current cursor position.
<Control>-w	Delete the text between the mark and the current cursor position, copying the deleted text to the cut buffer.
<Control>-k	"Kill" (delete) from the cursor to the end of the line, copying the deleted text into the cut buffer.
<Control>-y	"Yank" (paste) from the cut buffer into the command line, inserting it at the cursor location.

If the information being displayed is too long for your screen, the screen will show the "More" indication where the information breaks.

Table 1-5 shows the keystrokes for controlling the display of information in a "More" screen.

Table 1-5 Display options within a "More" screen

Type this:	To do this:
q Q	Exit "More"
<Space> f <Ctrl>+f	Scroll down one whole screen.
b <Ctrl>+b	Scroll up one whole screen
d <Ctrl>+d	Scroll down one-half screen.

Table 1-5 Display options within a “More” screen

Type this:	To do this:
u <Ctrl>+u	Scroll up one-half screen
<Enter> e <Ctrl>+e <Down Arrow>	Scroll down one line.
y <Ctrl>+y <Up Arrow>	Scroll up one line.
G	Scroll down to the bottom of the output.
g	Scroll up to the top of the output.
h	Display detailed help for “More”.

Filtering Command Output

The Vyatta system can pipe the output of commands into selected operating system shell commands to filter what is displayed on the console. Commands are piped into the filters using the vertical bar pipe operator (“|”).

Table 1-6 shows the pipe commands implemented for the Vyatta system.

Table 1-6 “Pipe” filter commands

Type this:	To do this:
count	Count occurrences.
match *pattern*	Show only text that matches the specified pattern.
more	Paginate output
no-match *pattern*	Show only text that does not match the specified pattern.
no-more	Don't paginate output.

Operational Commands

This section presents the following topics:

- Running Operational Commands
- Running an Operational Command in Configuration Mode

Running Operational Commands

Operational commands are executed in operational mode. The operational commands available to you can be displayed by entering **help** at the command prompt in operational mode.

Running an Operational Command in Configuration Mode

You can run an operational command without leaving configuration mode using the **run** command, as in the following example:

```
vyatta@R1# run show system processes summary
 20:45:46 up 1 day, 10:16,  3 users,  load average: 0.00, 0.00, 0.00
vyatta@R1#
```

Basic Commands for Using the CLI

Many basic commands for using the CLI are related to configuration. These commands have their own chapter: Chapter 2: Working with Configuration.

A great many shell commands are available in the operational mode of the CLI. Use command-line help to see shell commands available to you.

Feature-related commands available in operational mode are documented with the respective feature. The *Guide to Vyatta Documentation* can help you see how these features are organized with the Vyatta library of technical documentation.

This section describes the following commands.

Configuration Commands	
None	
Operational Commands	
copy file <from-file> to <to-file>	Copies files or directories.
delete file <file>	Deletes files or directories.
exit (operational)	Exits the system.
run	Runs an operational command without leaving configuration mode.
show file <file>	Displays information about files.

copy file <from-file> to <to-file>

Copies files or directories.

Syntax

copy file *from-file* **to** *to-file*

Command Mode

Operational mode.

Parameters

from-file	The source file or directory.
to-file	The destination file or directory.

Default

None.

Usage Guidelines

Use this command to copy a file or directory to a destination.

This command is optimized for configuration files and directories, in that command completion defaults to the **/config** directory of all known system images. For example, **running://config/** indicates the **/config** directory of the currently running system, and **test-image1://config/** indicates the **/config** directory of an image called "test-image1."If needed, however, any other location within the file system can be specified.

Files or directories can be copied on the local machine. Only files can be copied to and from remote machine using FTP, SCP, or TFTP.

NOTE *Use this command with caution, as its effects are not reversible.*

The following table shows how to specify different types of file locations.

Table 1-7 Specifying file locations

Location	Specification
FTP server	ftp://*user*:*passwd*@*host*/*file* where *user* is the user name on the host, *passwd* is the password associated with the user name, *host* is the host name or IP address of the FTP server, and *file* is the file, including the path. If you do not specify *user* and *passwd,* the system prompts you for them.
SCP server	scp://*user*:*passwd*@*host*/*file* where *user* is the user name on the host, *passwd* is the password associated with the user name, *host* is the host name or IP address of the SCP server, and *file* is the file, including the path. If you do not specify *user* and *passwd,* the system prompts you for them.
TFTP server	tftp://*host*/*file* where *host* is the host name or IP address of the TFTP server, and *file* is the file, including the path relative to the TFTP root directory.
The running (active) configuration	running://*path*/*file* where *path* is the path to the file, and *file* is the file.
A binary image	***image-name***://*path*/*file* where *image-name* is the name of a binary image, *path* is the path to the file, and *file* is the file.

Examples

Example 1-1 copies the contents of the **/config/x509/** directory on the currently running system to the **/config/x509/** directory of the TEST-IMAGE-1 image.

Example 1-1 Copying configuration files to another system image

```
vyatta@vyatta:~$ copy file running://config/auth/x509/ to
TEST-IMAGE-1://config/auth/x509/
sending incremental file list
created directory /live/image/boot/TEST-IMAGE-1/live-rw/config/x509
./
ca.crt
        1265 100%    0.00kB/s    0:00:00 (xfer#1, to-check=5/7)
crl.pem
         568 100%  554.69kB/s    0:00:00 (xfer#2, to-check=4/7)
key
```

```
        5626 100%    5.37MB/s    0:00:00 (xfer#3, to-check=3/7)
straylight-r1.crt
        3632 100%    3.46MB/s    0:00:00 (xfer#4, to-check=2/7)
straylight-r1.key
         891 100%  870.12kB/s    0:00:00 (xfer#5, to-check=1/7)
test.key
         401 100%  391.60kB/s    0:00:00 (xfer#6, to-check=0/7)

sent 12808 bytes  received 129 bytes  25874.00 bytes/sec
total size is 12383  speedup is 0.96vyatta@vyatta:~$
```

delete file <file>

Deletes files or directories.

Syntax

delete file *file*

Command Mode

Operational mode.

Parameters

file	The file or directory to delete, including the path.

Default

None.

Usage Guidelines

Use this command to delete a file or directory.

This command is optimized for configuration files and directories, in that command completion defaults to the **/config** directory of all known system images. For example, **running://config/** indicates the **/config** directory of the currently running system, and **test-image1://config/** indicates the **/config** directory of an image called “test-image1.”If needed, however, any other location within the file system can be specified.

NOTE *Use this command with caution as its effects are not reversible.*

Examples

Example 1-2 deletes the file **/config/user-data/xxx** on the currently running system.

Example 1-2 Deleting configuration user data

```
vyatta@vyatta:~$ delete file running://config/user-data/xxx
Do you want to erase the running://config/user-data/xxx file? (Y/N): y
File erased
vyatta@vyatta:~$
```

exit (operational)

Exits the system.

Syntax

exit

Command Mode

Operational mode.

Configuration Statement

None.

Parameters

None.

Default

None.

Usage Guidelines

Use this command in operational mode to exit from the system.

run

Runs an operational command without leaving configuration mode.

Syntax

run *command*

Command Mode

Configuration mode.

Configuration Statement

None.

Parameters

command	The operational command to be executed.

Default

None.

Usage Guidelines

Use this command to run an operational command without leaving configuration mode.

Examples

Example 1-3 executes the **show date** command (an operational command) from configuration mode.

Example 1-3 Running an operational command in configuration mode

```
vyatta@vyatta# run show date
Sun Dec 16 23:34:06 GMT 2007
vyatta@vyatta#
```

show file <file>

Displays information about files.

Syntax

show file *file*

Command Mode

Operational mode.

Parameters

file	The file or directory to display information for.

Default

None.

Usage Guidelines

Use this command to display information about the specified file or directory.

This command is optimized for configuration files and directories, in that command completion defaults to the **/config** directory of all known system images. For example, **running://config/** indicates the **/config** directory of the currently running system, and **test-image1://config/** indicates the **/config** directory of an image called "test-image1."If needed, however, any other location within the file system can be specified.

Different information will be displayed for various file types as shown in the following table.

Table 1-8 Types of information dispalyed for various file types

File type	Information displayed
Directory	Directory contents.
Text file	Information about the file as well as file contents.
Packet capture file (*.pcap)	Information about the file as well as file contents in the form of a packet capture from **tshark**.

Table 1-8 Types of information dispalyed for various file types

File type	Information displayed
Binary file	Information about the file as well as file contents in the form of a hex dump.

Examples

Example 1-4 shows the contents of the **/config/ips** directory on the currently running system.

Example 1-4 Displaying /config/ips

```
vyatta@vyatta:~$ show file running://config/ips
########### DIRECTORY LISTING ###########
total 30M
drwxr-sr-x 1 vyatta 4.0K Apr 20 08:38 etc/
drwxr-sr-x 1 vyatta 4.0K Mar  8 16:21 preproc_rules/
drwxrwsr-x 1 root   4.0K Apr 20 08:29 rules/
drwxr-sr-x 1 vyatta 4.0K Mar  8 16:11 so_rules/
-rwxrwxr-x 1 root      0 Mar 23 00:18 modify-sids
-rwxrwxr-x 1 root    30M Apr 11 16:00 snortrules-snapshot-latest.tar.gz
-rwxrwxr-x 1 vyatta    0 Apr  7 22:01 test
vyatta@vyatta:~$
```

Example 1-5 shows the partial contents of the **/config/config.boot** file on the currently running system.

Example 1-5 Displaying /config/config.boot

```
vyatta@vyatta:~$ show file running://config/config.boot
########### FILE INFO ###########
File Name: running://config/config.boot
Text File:
  Permissions: -rwxrwxr-x
  Owner:        root
  Size:         3.1K
  Modified:     Apr 21 18:51
  Description:  ASCII C program text

########### FILE DATA ###########
firewall {
    all-ping enable
    broadcast-ping disable
    conntrack-expect-table-size 4096
    conntrack-hash-size 4096
    conntrack-table-size 32768
...
```

Example 1-6 shows the partial contents of the **/tmp/test1.pcap** file on the currently running system.

Example 1-6 Displaying /tmp/test1.pcap

```
vyatta@vyatta:~$ show file running://tmp/test1.pcap
########### FILE INFO ###########
Binary File:
  Permissions: -rw-------
  Owner:         root
  Size:          35K
  Modified:      Apr 24 19:41
  Description:   tcpdump capture file (little-endian) - version 2.4
(Ethernet, capture length 65535)

########### FILE DATA ###########
1   0.000000 192.168.56.101 -> 192.168.56.1 SSH Encrypted response packet
len=128
  2   0.000155 192.168.56.1 -> 192.168.56.101 TCP 54566 > ssh [ACK] Seq=1
Ack=129 Win=1002 Len=0 TSV=186250939 TSER=21591709
  3   0.259966 192.168.56.101 -> 192.168.56.1 SSH Encrypted response packet
len=48
  4   0.260216 192.168.56.1 -> 192.168.56.101 TCP 54566 > ssh [ACK] Seq=1
Ack=177 Win=1002 Len=0 TSV=186251199 TSER=21591735
...
```

Example 1-7 shows the partial contents of the **/config/r1.tar** file on the currently running system.

Example 1-7 Displaying /config/r1.tar

```
vyatta@vyatta:~$ show file running://config/r1.tar
File Name: running://config/r1.tar
Binary File:
  Permissions: -rwxrwxr-x
  Owner:         vyatta
  Size:          20K
  Modified:      Feb 6 23:09
  Description:   POSIX tar archive (GNU)
```

```
########### FILE DATA ###########
00000000  72 31 2f 00 00 00 00 00  00 00 00 00 00 00 00 00
|r1/.............|
00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
|................|
*
...
```

Chapter 2: Working with Configuration

This chapter describes configuration management utilities on the Vyatta system.

This chapter presents the following topics:

- Configuration Basics
- Changing Configuration Information
- Managing System Configuration
- Safeguarding Remote Configuration Sessions
- Commands for Working with Configuration

Configuration Basics

This section presents the following topics:

- Terminology
- Location of Configuration Information
- Configuration Hierarchy
- Entering and Exiting Configuration Mode
- Navigating in Configuration Mode
- Viewing Configuration
- Viewing Configuration from Operational Mode

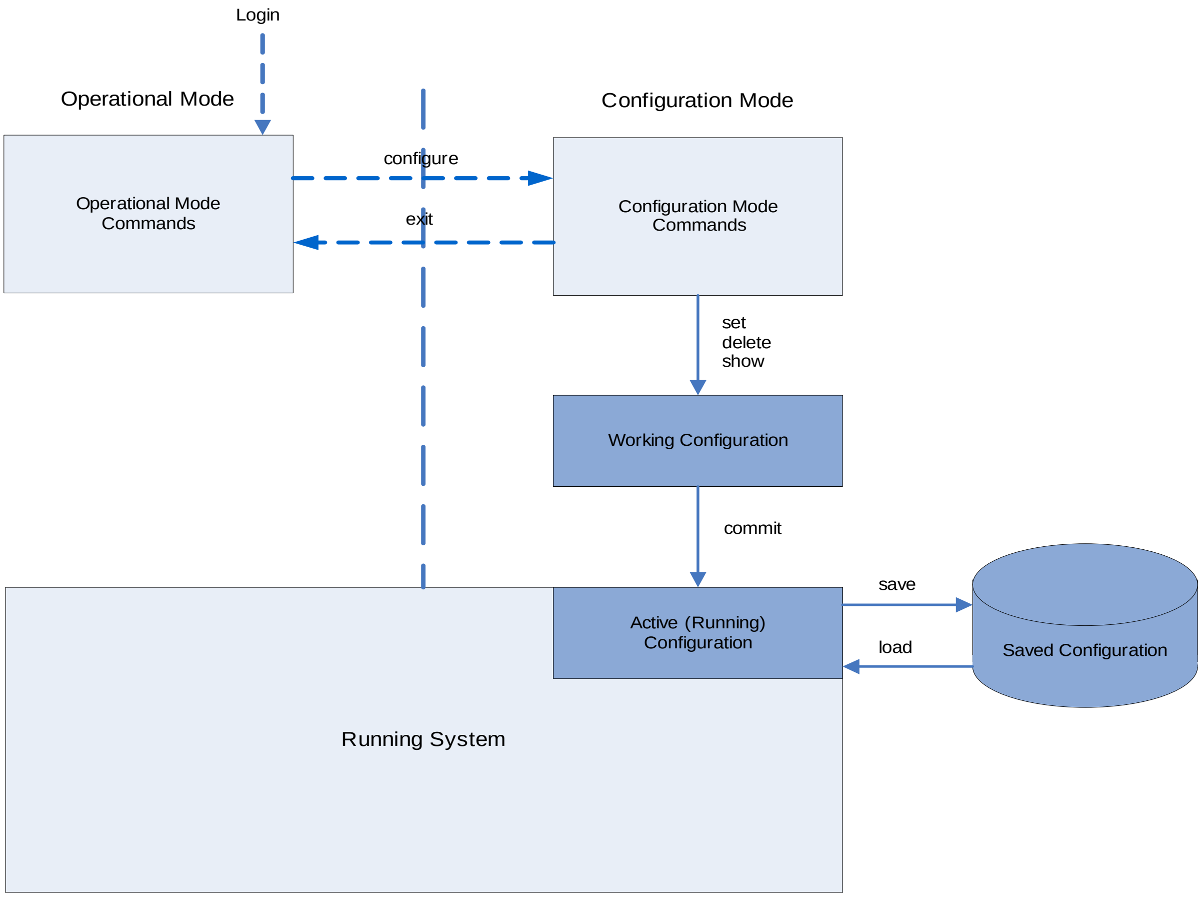
Terminology

Several versions of system configuration information exist on the system at a given time:

- **Active or "running" configuration**. This is the configuration that is actually loaded and being used by the system.
- **Working configuration**. When you enter configuration mode and make configuration changes, changes remain in working configuration until you commit the changes, at which time the configuration becomes active or running.
- **Saved or "boot" configuration**. If you save configuration (using the **save** command), it is saved to the file **config.boot** in the **/config** directory of the local system. When you reboot, the system reads and loads the configuration from **config.boot**.

Figure 2-1 shows configuration states possible in the Vyatta CLI.

Figure 2-1 CLI configuration states



Location of Configuration Information

Boot configuration is stored in **config.boot** in the **/config** directory. In addition to the **config.boot** file, the **/config** directory has a number of subdirectories, each with a specific function, as follows:

- **archive**. This directory stores archived versions of configuration.
- **auth**. This directory stores security certificates referenced in the configuration tree; for example, OpenVPN certificates, IPsec certificates, and RSA/IPsec keys. You can add additional structure to this directory—for example, to store X.509 certificates, you can add an **/auth/x509** directory. To ensure smooth upgrades, and to preserve this kind of information across upgrades, ensure that any certificate file you reference within a configuration node is stored here.
- **ips**. This directory stores Intrusion Prevention System (IPS) rules and other IPS data on which content-inspection configuration depends.

- **scripts**. This directory stores scripts referenced from within the configuration nodes; for example, VRRP transition scripts, the WAN load-balancing hook, and so on. To ensure smooth upgrades, and to preserve this kind of information across upgrades, ensure that any script file you reference within a configuration node is stored here.
- **support**. This directory stores system information generated by the **show tech-support save** command.
- **url-filtering.** This directory stores the URL-filtering database and files on which web proxy and URL filtering depend.
- **user-data**. This directory stores user-generated scripts and user data. To ensure smooth upgrades, ensure that any user data that needs to be preserved across upgrades is stored here.

You can freely use the **user-data** subdirectory to store any of your own information you want to preserve across system upgrades. The other subdirectories, including **auth** and **scripts**, contain information on which system operation relies, and you should only make changes to them with great care.

Configuration Hierarchy

Vyatta system configuration is organized as a hierarchy of configuration statements, with a hierarchical tree of *nodes* similar to the directory structure on a UNIX file system. There are three kinds of statements:

- Configuration nodes. These can be either:
    - Single nodes (just one instance can be created; for example, the **rip protocol** node)
    - Multi-nodes (more than one instance can be created; for example, **address** nodes)
- Attribute statements. These set the values or characteristics for parameters within a node.

From the system’s point of view, a configuration node is different from a simple configuration attribute statement. A configuration *attribute statement* takes the form *attribute value*, as in the following example.

```
protocol-version v2
```

A configuration *node* always has an enclosing pair of braces, which may be empty, as in the following example:

```
service {
    https {}
}
```

or non-empty, as in the following example:

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

Entering and Exiting Configuration Mode

To enter configuration mode, use the **configure** command in operational mode.

```
Entering configuration mode

vyatta@vyatta:~$ configure
vyatta@vyatta#
```

Once in configuration mode, the command prompt changes from this:

```
user@host:~$
```

to this:

```
user@host#
```

To exit configuration mode, use the **exit** command from the top level of configuration.

If you have changed configuration, you must either **commit** changes or discard them using the **exit discard** command.

Navigating in Configuration Mode

You can tell where you are in the configuration tree by the **[edit]** prompt, which is context-sensitive.

At the top of the configuration tree, the [**edit**] prompt displays like this:

```
[edit]
```

When you are in another location, the edit prompt displays your location by showing the node hierarchy in order, like this:

```
[edit protocols bgp 65537]
```

Table 2-1 shows the commands for navigating in configuration mode.

Table 2-1 Commands for navigating in configuration mode

Command	Result
edit *config-node*	Navigates to the specified configuration node for editing. The node must already be created the configuration committed.
exit	Jumps to the top of the configuration tree. If you are already at the top of the configuration tree, exit from configuration mode and return to operational mode.
top	Jumps to the top of the configuration tree.
up	Moves up one node in the configuration tree.

Using the **edit** command lets you navigate to the part of the hierarchy that you are interested in and execute commands relative to your location. This saves typing if you need to work on a particular part of the configuration hierarchy.

The following example navigates to the configuration node for the Ethernet interface eth2. Once you have navigated to the node, you can show configuration directly without specifying the full path.

```
vyatta@R1# edit interfaces ethernet eth2
[edit interfaces ethernet eth2]
vyatta@R1# show
 hw-id 00:13:46:e6:f6:87
[edit interfaces ethernet eth2]
vyatta@R1#
```

Viewing Configuration

Use the **show** command in configuration mode to display configuration. You can restrict the display to a particular node by specifying the path to the node.

The following example shows configuration for all configured interfaces.

```
vyatta@R1# show interfaces
    ethernet eth0 {
        address 10.1.0.62/24
        hw-id 00:40:63:e2:e4:00
    }
    ethernet eth1 {
        address 172.16.234.23/25
        hw-id 00:40:63:e2:e3:dd
        vrrp {
            virtual-address 172.16.99.99
            vrrp-group 20
        }
    }
    loopback lo {
    }
}
```

The following example shows configuration only for the Ethernet interface eth0.

```
vyatta@R1# show interfaces ethernet eth0
 address 10.1.0.62/24
 hw-id 00:40:63:e2:e4:00
```

When the display is too large for one screen, it stops with one screen displayed. In this case:

- Press <Enter> to display the next line.
- Press <space> to display the next screen.
- Press q to interrupt the display and return to the command prompt.

Viewing Configuration from Operational Mode

You can display configuration information without leaving operational mode using the **show configuration** command, as in the following example:

```
vyatta@R1:~$ show configuration
interfaces {
     ethernet eth0 {
         address 192.168.1.77/24
         hw-id 00:0c:29:68:b3:9f
     }
     ethernet eth1 {
         hw-id 00:0c:29:68:b3:a9
     }
     loopback lo {
     }
}
service {
     ssh {
     }
}
system {
     gateway-address 192.168.1.254
     host-name R1
     login {
         user vyatta {
             authentication {
                 encrypted-password ****************
:
```

Changing Configuration Information

This section presents the following topics:

- Adding or Modifying Configuration
- Deleting Configuration
- Committing Configuration Changes
- Discarding Configuration Changes
- Cloning a Configuration Node
- Renaming a Configuration Node
- Adding Comments to a Configuration Node
- Deleting Comments from a Configuration Node

Adding or Modifying Configuration

Add new configuration by creating a configuration node, using the **set** command in configuration mode. Modify existing configuration using the **set** command in configuration mode, as in the following example:

```
vyatta@R1# set interfaces ethernet eth2 address 192.168.1.100/24
vyatta@R1#
```

Then use the **show** command to see the change:

```
vyatta@R1# show interfaces ethernet eth2
+address 192.168.1.100/24
 hw-id 00:13:46:e6:f6:87
vyatta@R1#
```

Note the "+" in front of the new statement. This shows that this statement has been added to the configuration but the change is not yet committed. The change does not take effect until configuration is committed using the **commit** command.

You can modify configuration from the root of the configuration tree or use the **edit** command to navigate to the part of the tree where you want to change or add.

The configuration tree is nearly empty when you first start up, except for a few automatically configured nodes. You must create a node for any functionality you want to configure on the system. When a node is created, any default values that exist for its attributes are applied to the node.

Deleting Configuration

Use the **delete** command to delete a configuration statement or a complete configuration node, as in the following example:

```
vyatta@R1# delete interfaces ethernet eth2 address 192.168.1.100/24
```

Then use the **show** command to see the change:

```
vyatta@R1# show interfaces ethernet eth2
-address 192.168.1.100/24
 hw-id 00:13:46:e6:f6:87
```

Note the “-” in front of the deleted statement. This shows that this statement has been deleted from the configuration but the change is not yet committed. The change does not take effect until configuration is committed using the **commit** command.

Some configuration nodes are mandatory; these cannot be deleted. Some configuration nodes are mandatory, but have default values; if you delete one of these nodes, the default value is restored.

Committing Configuration Changes

In the Vyatta System, configuration changes do not take effect until you commit them using the **commit** command.

```
vyatta@R1# commit
```

Lines with uncommitted changes in them are flagged as follows:

- “>” to indicate the line has been modified
- “+” to indicate the line has been added
- “-” to indicate the line has been deleted

Once you commit the changes, the flag disappears, as in the following example.

```
vyatta@R1# show interfaces ethernet eth2
-address 192.168.1.100/24
 hw-id 00:13:46:e6:f6:87
vyatta@R1# commit
vyatta@R1# show interfaces ethernet eth2
 hw-id 00:13:46:e6:f6:87
```

Discarding Configuration Changes

You cannot exit from configuration mode with uncommitted configuration changes; you must either commit the changes or discard them. If you don’t want to commit the changes, you can discard them using the **exit discard** command.

```
vyatta@R1# exit
Cannot exit: configuration modified.
Use 'exit discard' to discard the changes and exit.
vyatta@R1# exit discard
```

```
vyatta@R1:~$
```

Cloning a Configuration Node

To save time entering information, you can copy, or clone, a configuration multi-node. Configuration multi-nodes (that is, nodes that allow for multiple instances) are distinguished from one another by their identifiers. For example, firewall and NAT rules have numbers; firewall rule sets have names, IPsec VPN proposals have names, and system users have user IDs.

To clone a configuration node, navigate to the point in the configuration hierarchy just above the node that you want to copy. Then use the **copy** command to change the identifier. An example is provided on page 60.

Renaming a Configuration Node

One thing you can’t do with the **set** command is change the identifier of a node for which there can be multiple instances (a “multi-node”), such as a DNS server or an IP address for an interface. However, if a multi-node has an incorrect identifier, you can change the identifier using the **rename** command.

To rename a configuration node, navigate to the point in the configuration hierarchy just above the node that you want to rename. Then use the **rename** command to change the identifier. An example is provided on page 76.

Adding Comments to a Configuration Node

In complicated configurations, it can be helpful to add comments to help you remember or help another person understand what a particular configuration does. The Vyatta system allows you to annotate your configuration using the **comment** command.

To annotate configuration, specify the configuration node and the text to be added as a comment, as in the following example:

```
vyatta@vyatta# comment interfaces ethernet eth2 “Connection to DMZ”

vyatta@vyatta# show interfaces ethernet eth2
/* Connection to DMZ */
   eth2 {
       address 192.168.22.22/24
       hw-id 3a:9b:98:c3:46:8c
```

```
        mtu 1500
    }
```

Deleting Comments from a Configuration Node

To remove a comment using the **comment** command, specify the configuration node and an empty string, as in the following example:

```
vyatta@vyatta# comment interfaces ethernet eth2 “”

vyatta@vyatta# show interfaces ethernet eth2
    eth2 {
        address 192.168.22.22/24
        hw-id 3a:9b:98:c3:46:8c
        mtu 1500
    }
```

Managing System Configuration

This section presents the following topics:

- Saving the Running Configuration
- Loading a Saved Configuration
- Booting from a Saved Configuration File
- Merging Saved and Running Configurations
- Archiving Configuration Versions on Commit
- Comparing Configuration Versions
- Rolling Back to a Previous Version
- Cloning Configuration Across System Images
- Performing File Operations on Configuration Files and Directories

Saving the Running Configuration

Save the running configuration using the **save** command in configuration mode. By default, configuration is saved to the file **config.boot** in the standard configuration directory.

- For hard disk installs, the configuration directory is **/config**.
- For installations running off a LiveCD, the configuration directory is **/media/floppy/config**.

```
vyatta@R1# save
Saving configuration to '/config/config.boot'...
Done
vyatta@R1#
```

You can save configuration to a different location by specifying a different file name.

```
vyatta#R1 save testconfig
Saving configuration to '/config/testconfig'...
Done
vyatta@R1#
```

You can also save a configuration file to a location path other than the standard configuration directory by specifying a different path. You can save to a hard drive, compact Flash, or USB device.

Note that the **save** command writes only committed changes. If you try to save uncommitted changes the system warns you that it is saving only the committed changes.

Table 2-2 shows the syntax for file specifications you can use in file paths when you save files in configuration mode.

Table 2-2 Specifying locations for the configuration file

Location	Specification
An absolute path	Use standard UNIX file specification.
A relative path	Specify the path name relative to the location configured for the config-directory parameter of the rtrmgr configuration node.
TFTP server	Use the following syntax for *file-name*: tftp://*ip-address*/*config-file* where *ip-address* is the IP address of the TFTP server, and *config-file* is the configuration file, including the path relative to the TFTP root directory.

Table 2-2 Specifying locations for the configuration file

Location	Specification
FTP server	Use the following syntax for *file-name*: ftp://*ip-address*/*config-file* where *ip-address* is the IP address of the FTP server, and *config-file* is the configuration file, including the path. If you use FTP, you will be prompted for a user name and password.
HTTP server	Use the following syntax for file-name: http://*ip-address*/*config-fil*e where *ip-address* is the IP address of the HTTP server, and *config-file* is the configuration file, including the path.

If you are running the system from LiveCD, configuration can be saved only to floppy disk. If you do not save your running configuration to floppy disk, any changes you have made will be lost after reboot.

Before you can save configuration to a floppy disk, you must initialize the floppy disk using the **init-floppy** command in operational mode.

Loading a Saved Configuration

To load a previously saved configuration use the **load** command in configuration mode. By default, the system reads the file from the standard configuration directory.

- For hard disk installs, the configuration directory is **/config**.
- For installations running off a LiveCD, the configuration directory is **/media/floppy/config**.

```
vyatta@R1# load testconfig
Loading config file /config/testconfig...

Load complete.  Use 'commit' to make changes active.
[edit]
vyatta@R1#
```

A loaded configuration needs to then be **commit**ted to become the active configuration.

Booting from a Saved Configuration File

If you want the file to be automatically read the next time the system starts, you must save it as **config.boot** in the **/config** directory. By default:

- For hard disk installs the configuration directory is **/config**.
- For installations running off of a LiveCD, the configuration directory is is **/media/floppy/config**.

Merging Saved and Running Configurations

You can merge a saved configuration with the active (running) configuration using the **merge** command. An example is provided in Example 2-16.

The process of merging adds new configuration entries and applies any modifications to existing active entries to produce a new working configuration. This merged configuration must be committed before it becomes the active configuration.

Configuration can be loaded from a hard disk (including a Flash disk or USB device), a TFTP server, an FTP server, an SCP server, or an HTTP server. Note that you cannot load an empty configuration file; the configuration file must contain at least one configuration node.

Archiving Configuration Versions on Commit

The system automatically archives the configuration whenever you commit a configuration change. The new, committed configuration version is saved to **config.boot** in **/config**. The old **config.boot** file is saved to the **/config/archive** directory under the name **config.boot**.*timestamp*, where *timestamp* is the time the file was saved in the format *YYYY-MM-DD-hhmmss*.

By default, the system maintains 20 versions of configuration in the archive. You can change the number of versions maintained in the archive using the **system config-management commit-revisions** command.

You can also direct the system to save configuration versions to a remote location whenever configuration is committed, using the **system config-management commit-archive** command. FTP, SCP, and TFTP destinations are supported.

Rolling Back to a Previous Version

You can roll back system configuration to any archived version using the **rollback** command. To see the list of configuration file revisions available, use the **show system commit** operational mode command.

Comparing Configuration Versions

You can compare two versions of configuration using the **show system commit** and **compare** commands. Table 2-3 summarizes options for comparing configuration versions.

Table 2-3 Commands for comparing configuration versions

Use This Command	In Order to:
Configuration Commands	
compare	See the difference between the working and active configuration.
compare *n*	See the difference between the working configuration and revision *n*.
compare *n m*	See the difference between revision *n* and revision *m*.
Operational Commands	
show system commit	See the summary of commits.
show system commit file *n*	See the full configuration at revision *n*.
show system commit file *n* compare *m*	See the difference between revision *n* and revision *m*.
show system commit diff *n*	See what changed in a given commit (between revision *n* and revision *n* + 1). This command is equivalent to the command **show system file** *n* **compare** *n*+1.

Cloning Configuration Across System Images

You can copy the **/config** directory from one image to another using the **clone system config** command.

This command copies the **/config** directory from the running configuration (or another specified configuration) to the **/config** directory of another specified image. You should use this command with caution, as it overwrites the entire **/config** directory of the destination image and its effects are not reversible.

Performing File Operations on Configuration Files and Directories

The Vyatta system supports several general file operation commands that are optimized for working with image and configuration files. They are the **show file**, **copy file**, and **delete file** commands. These commands are documented in Chapter 1: Using the CLI.

These commands are optimized for configuration files and directories, because command completion defaults to the **/config** directory of all known system images. For example, **running://config/** indicates the **/config** directory of the currently running system, and **test-image1://config/** indicates the **/config** directory of an image called “test-image1.”If needed, however, any other location within the file system can be specified.

Safeguarding Remote Configuration Sessions

One of the risks in configuration sessions over a remote connection is that a configuration mistake will be made that causes permanent loss of remote access to the device. An example is a mistake in configuring the IP address of the management port. To safeguard against critical configuration mistakes, you can use the **commit-confirm** command to set the Vyatta system to require you to confirm configuration changes.

After entering the sensitive configuration, but before committing it, enter the **commit-confirm** command, specifying the allowed confirmation interval. If the commit goes without incident, enter the **confirm** command to confirm configuration. If confirmation within the required interval does not occur—for example, because it is impossible due to the session dropping—the remote system reboots to the last known good configuration.

Commands for Working with Configuration

This section presents the following commands.

Configuration Commands	
Configuration Basics	
edit	Navigates to a subnode in the configuration tree for editing.
exit (configuration)	Navigates up one level of use.
show	Displays configuration information in configuration mode.
top	Moves to the top level of the configuration hierarchy.
up	Navigates up one level in the configuration tree.
Changing Configuration Information	
comment	Adds a comment as an annotation to a configuration node
commit	Applies any uncommitted configuration changes.
copy	Allows you to copy, or clone, a configuration node.
delete	Deletes a configuration node.
discard	Discards any uncommitted configuration changes.
rename	Allows you to change the identifier of a named configuration node.
set	Creates a new configuration node, or modifies a value in an existing configuration node.
Managing System Configuration	
clone system config <dest-image-name>	Clones the configuration directory of one image to another image.
compare	Allows you to compare two sets of configuration information.
load	Loads a saved configuration.
merge	Merges a saved configuration with the active (running) configuration.
rollback <rev-num>	Allows you to roll back configuration to a specific revision.
save	Saves the running configuration to a file.

system config-management commit-archive location <location>	Enables archiving of configuration to the specified location every time a change is committed.
system config-management commit-revisions <revisions>	Specifies the number of configuration revisions to store locally.
Safeguarding Remote Configuration Sessions	
commit-confirm <minutes>	Reboots to the last known good configuration unless confirmation is supplied for the commit.
confirm	Allows you to confirm to the system that a commit was successful.
Operational Commands	
configure	Enters configuration mode.
show configuration (operational)	Displays system configuration from operational mode.
show system commit	Displays a summary of configuration file revisions.
show system commit diff <rev-num>	Compares adjacent configuration file revisions.
show system commit file <rev-num>	Displays a specific configuration file revision.

The following commands are optimized for working with files across images.

Related Commands Documented Elsewhere	
copy file <from-file> to <to-file> delete file <file> show file <file>	These commands allow you to perform general file management tasks, but use image-relative completion to make it easy to work with files in different images.
show log image <image-name>	This command allows you to view log files across multiple images.

clone system config <dest-image-name>

Clones the configuration directory of one image to another image.

Syntax

clone system config *dest-image-name* [from *source-image-name*]

Command Mode

Operational mode.

Parameters

dest-image-name	The name of the image that the configuration directory will be copied to.
source-image-name	Optional. The name of the image that the configuration directory will be copied from.

Default

The configuration directory is copied from the running system.

Usage Guidelines

Use this command to copy the **/config** directory from one image to another. By default the source image is the currently running image.

This command is equivalent to the command **copy file running://config/ to** *dest-image-name*:**//config/**

NOTE *Use this command with caution as its overwrites the entire /config directory of the destination image and its effects are not reversible.*

Command completion displays all valid system images. It is not possible to clone to the running image or the “disk-install” image.

Examples

Example 2-1 copies the contents of the **/config** directory of the currently running system to the **/config** directory of the TEST-IMAGE-1 image.

Example 2-1 Copying /config from the running system to another system image

```
vyatta@vyatta:~$ clone system config TEST-IMAGE-1
WARNING: This is a destructive copy of the /config directories
This will erase all data in the TEST-IMAGE-1://config directory
This data will be replaced with the data from running://
Do you wish to continue? (Y/N): y
config/
config/.vyatta_config
...
```

Example 2-2 copies the contents of the /config directory of the TEST-IMAGE-2 system to the /config directory of the TEST-IMAGE-1 image.

Example 2-2 Copying /config/ from the running system to another system image

```
vyatta@vyatta:~$ clone system config TEST-IMAGE-1 from TEST-IMAGE-2
WARNING: This is a destructive copy of the /config directories
This will erase all data in the TEST-IMAGE-1://config directory
This data will be replaced with the data from TEST-IMAGE-2
Do you wish to continue? (Y/N): y
sending incremental file list
config/
config/.vyatta_config
...
```

comment

Adds a comment as an annotation to a configuration node

Syntax

comment *config-node comment-text*

Command Mode

Configuration mode.

Configuration Statement

None.

Parameters

config-node	The configuration node to be commented, including the full path through the configuration hierarchy to the node, separated by spaces.
comment-text	The text to be added as a comment to the configuration. A null string, represented by two double quotes (“”), is used to delete an existing comment.

Default

None.

Usage Guidelines

Use this command to add a comment to the configuration or to remove an existing comment.

Comments are placed directly above the node being commented and are enclosed with “/*” and “*/” as opening and closing delimiters, respectively.

To remove an existing comment, specify an empty string enclosed in double quote marks (“”) as the comment text.

Examples

Example 2-3 adds the comment “Connection to DMZ” to the configuration for the eth2 Ethernet interface.

Example 2-3 Adding comments to configuration

```
vyatta@vyatta# comment interfaces ethernet eth2 “Connection to DMZ”

vyatta@vyatta# show interfaces ethernet
   eth0 {
       address 192.168.1.82/24
       hw-id 2a:91:0b:00:f6:9e
   }
   eth1 {
       duplex auto
       hw-id b6:cc:6a:95:22:b2
   }
/* Connection to DMZ */
   eth2 {
       address 192.168.22.22/24
       hw-id 3a:9b:98:c3:46:8c
       mtu 1500
   }
```

Example 2-4 removes a comment from a configuration.

Example 2-4 Removing a comment from configuration

```
vyatta@vyatta# show interfaces ethernet
   eth0 {
       address 192.168.1.82/24
       hw-id 2a:91:0b:00:f6:9e
   }
   eth1 {
       duplex auto
       hw-id b6:cc:6a:95:22:b2
   }
/* Connection to DMZ */
   eth2 {
       address 192.168.22.22/24
       hw-id 3a:9b:98:c3:46:8c
       mtu 1500
   }
vyatta@vyatta# comment interfaces ethernet eth2 “”
```

```
vyatta@vyatta# show interfaces ethernet
   eth0 {
       address 192.168.1.82/24
       hw-id 2a:91:0b:00:f6:9e
   }
   eth1 {
       duplex auto
       hw-id b6:cc:6a:95:22:b2
   }
   eth2 {
       address 192.168.22.22/24
       hw-id 3a:9b:98:c3:46:8c
       mtu 1500
   }
```

commit

Applies any uncommitted configuration changes.

Syntax

commit

Command Mode

Configuration mode.

Configuration Statement

None.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to apply changes to configuration.

When you add configuration to the system, modify existing configuration, or delete configuration from the system, the changes you make must be committed before they take effect. To do this, you issue the **commit** statement.

If you try to exit or quit from configuration mode while there are still uncommitted configuration changes, the system will give you a warning. You will not be able to exit from configuration mode until you either commit the changes by issuing the **commit** statement, or you discard the changes using the **exit discard** statement (see page 67).

Until a configuration change is committed, the system marks the change when displaying the information.

Committing information can take time, depending on the complexity of the configuration and how busy the system is. Be prepared to wait for several seconds for the system to complete committing the information.

If two or more users are logged on to the system in configuration mode and one user changes the configuration, the other user(s) will receive a warning.

Examples

Example 2-5 shows an uncommitted deletion which is then committed. In this example, note how the uncommitted deletion is flagged with a minus sign ("-"), which disappears after the change is committed.

Example 2-5 Committing configuration changes

```
vyatta@vyatta# show interfaces ethernet eth2
-address 192.168.1.100/24
 hw-id 00:13:46:e6:f6:87
vyatta@vyatta# commit
vyatta@vyatta# show interfaces ethernet eth2
 hw-id 00:13:46:e6:f6:87
```

commit-confirm <minutes>

Reboots to the last known good configuration unless confirmation is supplied for the commit.

Syntax

commit-confirm *minutes* [**comment** *comment*]

Command Mode

Configuration mode.

Configuration Statement

None.

Parameters

minutes	The time, in minutes, to wait for confirmation to be supplied. If this interval expires, the system reboots to the last known good configuration.
comment	A comment to appear in the configuration file revision history. The format is a string enclosed in double quotes.

Default

None.

Usage Guidelines

Use this command to set the system to require confirmation of a configuration commit.

This operation is useful when making configuration changes over a remote connection that could cause you to be unable to connect back into the system. An example is accidentally changing the IP address of the management port.

After making the sensitive configuration change, but before committing the change, enter the commit-confirm command, specifying the confirmation interval. Commit the change. If the commit completes without incident, confirm the commit by entering the confirm command.

If you are unable to confirm by the time the confirmation interval expires—for example, because your session drops—the system automatically reboots to the configuration that was in effect prior to the configuration change. This restores remote access.

Examples

Example 2-6 shows the **commit-confirm** command being used.

Example 2-6 Showing the **commit-confirm** command rebooting a system

```
vyatta@R1# delete interfaces ethernet eth1 address
[edit]
vyatta@R1# commit-confirm 1 comment “interface deleted”
commit confirm will be automatically rebooted in 1 minutes unless
confirmed
Proceed? [confirm][y]

*** At this point the remote session drops since the address for the
interface was deleted. ***

Connection closed by foreign host.

*** After a couple minutes the remote session is retried ***

Test@host1:~$ telnet 172.16.117.2
Trying 172.16.117.2...
Connected to 172.16.117.2.
Escape character is '^]'.

Welcome to Vyatta
R1 login: vyatta
Password:
Last login: Tue Nov 16 08:52:37 PST 2010 on pts/0
Linux R1 2.6.35-1-586-vyatta-virt #1 SMP Thu Nov 11 18:16:31 PST 2010 i686
Welcome to Vyatta.
This system is open-source software. The exact distribution terms for
each module comprising the full system are described in the individual
files in /usr/share/doc/*/copyright.

*** After successfully reconnecting, we look at the revision history ***

vyatta@R1:~$ show system commit
0   2010-11-16 11:08:01 by root via cli
    rollback/reboot
1   2010-11-16 11:07:56 by vyatta via cli
    interface deleted
```

```
2   2010-11-16 11:05:57 by vyatta via cli
    enable commit-revs
3   2010-11-16 11:05:56 by root via cli
    baseline
```

compare

Allows you to compare two sets of configuration information.

Syntax

compare [[*rev-num1*] *rev-num*]

Command Mode

Configuration mode.

Configuration Statement

None.

Parameters

rev-num	A configuration file revision to be compared.
rev-num1	A configuration file revision to be compared.

Default

When used with no option, the working and active (“running”) configuration are compared. When only one revision number is specified, the system compares the working configuration to the specified revision.

Usage Guidelines

Use this command to compare two configurations while in configuration mode.

You can see the list of configuration file revisions using the show system commit operational mode command (use “run show system commit” from configuration mode).

Examples

Example 2-7 shows the working and active configurations being compared on R1.

Example 2-7 Displaying the comparison of the working and active configurations

```
vyatta@R1# compare
@@ -71,11 +71,11 @@
             speed 9600
         }
     }
-    flow-accounting {
-        interface eth0
-        interface eth1
-        syslog-facility daemon
-    }

     host-name R1
vyatta@R1#
```

Example 2-8 shows configuration revisions 1 and 2 being compared on R1.

Example 2-8 Displaying the comparison of configuration revisions 1 and 2

```
vyatta@R1# compare 1 2
@@ -84,6 +84,9 @@
       }
   }
   system {
+      config-management {
+          commit-revisions 20
+      }
       console {
vyatta@R1#
```

configure

Enters configuration mode.

Syntax

configure

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to enter configuration mode from operational mode. In configuration mode, you can add, delete, and modify configuration information.

When you are in configuration mode, the command prompt changes to mark the change in command mode.

Examples

Example 2-9 shows the system’s response to entering configuration mode. In this example, notice how the command prompt changes when the user enters configuration mode.

Example 2-9 Entering configuration mode

```
vyatta@vyatta:~$ configure
vyatta@vyatta#
```

confirm

Allows you to confirm to the system that a commit was successful.

Syntax

confirm

Command Mode

Configuration mode.

Configuration Statement

None.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to confirm successful configuration change after requiring commit confirmation.

For configuration changes that carry some risk of causing loss of access to a system, you can direct the system to require commit confirmation by using the commit-confirm <minutes> command. This command sets the system to wait for confirmation that a configurationhas succeeded.

Issuing this command within the specified commit-confirm interval causes the configuration change to be accepted. If confirmation is not provided by issuing this command, the system reboots to the last known good configuration.

Examples

Example 2-7 shows the **confirm** command being used.

Example 2-10 Showing the **confirm** command confirming the commit

```
vyatta@R1# commit-confirm 1
commit confirm will be automatically rebooted in 1 minutes unless
confirmed
Proceed? [confirm][y]
vyatta@R1:~$ confirm
[edit]
vyatta@R1:~$
```

copy

Allows you to copy, or clone, a configuration node.

Syntax

copy *from-config-node* **to** *to-config-node*

Command Mode

Configuration mode.

Configuration Statement

None.

Parameters

from-config-node	The configuration node to be copied. The format is a series of space-separated tokens representing the path through the configuration hieratchy to the node to be renamed; for example, **firewall name RULE-SET-1 rule 10**.
to-config-node	The configuration node to be created. The format is a series of space-separated tokens representing the path through the configuration hieratchy y to the new node; for example, **firewall name RULE-SET-1 rule 20**.

Default

None.

Usage Guidelines

Use this command to make a copy, or clone, of a configuration subnode.

To make specifying the configuration subnode easier, use this command in conjunction with the **edit** command. Use the **edit** command to navigate to the appropriate place in the configuration hierarchy, then copy the appropriate subnode.

If you show configuration before it is committed, you will see the copied statement flagged with a plus sign (“+”); this flag disappears after the configuration change is committed.

Examples

Example 2-11 shows a firewall rule being copied.

Example 2-11 Cloning configuration subnodes

```
vyatta@vyatta# show firewall
 name xxx {
    rule 10 {
       action accept
    }
 }
vyatta@vyatta# edit firewall name RULE-SET-1
[edit firewall name RULE-SET-1]
vyatta@vyatta# copy rule 10 to rule 20
[edit firewall name RULE-SET-1]
vyatta@vyatta# commit
[edit firewall name RULE-SET-1]
vyatta@vyatta# show
 rule 10 {
    action accept
 }
 rule 20 {
    action accept
 }
[edit firewall name RULE-SET-1]
vyatta@vyatta# top
```

delete

Deletes a configuration node.

Syntax

delete *config-node*

Command Mode

Configuration mode.

Configuration Statement

None.

Parameters

config-node	The configuration node to be deleted, including the full path, separated by spaces, through the configuration hierarchy to the node.

Default

None.

Usage Guidelines

Use this command to delete a part of configuration. To do this, you delete the appropriate subnode of a configuration node.

If you show configuration before it is committed, you will see the deleted statement flagged with a minus sign ("-"); the statement disappears after the configuration change is committed.

Some configuration nodes and statements are mandatory; these nodes or statements cannot be deleted. Some configuration statements are mandatory but have default values; if you delete one of these statements, the default value is restored.

Examples

Example 2-12 deletes a DNS server from system configuration.

Example 2-12 Deleting configuration

```
vyatta@vyatta# show system name-server <Tab>
10.0.0.30  10.0.0.31  10.0.0.32
vyatta@vyatta# delete system name-server 10.0.0.32
vyatta@vyatta# show system name-server <Tab>
10.0.0.30  10.0.0.31
```

discard

Discards any uncommitted configuration changes.

Syntax

discard

Command Mode

Configuration mode.

Configuration Statement

None.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to discard all uncommitted changes to configuration.

Examples

Example 2-13 shows an uncommitted deletion and an uncommitted addition which are then discarded. In this example, note how the uncommitted deletion (flagged with a minus sign "-") and the uncommitted addition (flagged with a plus sign "+"), disappear after the **discard** command is invoked.

Example 2-13 Discarding configuration changes

```
vyatta@vyatta# show interfaces ethernet eth2
-address 192.168.1.100/24
+address 192.168.1.101/24
 hw-id 00:13:46:e6:f6:87
vyatta@vyatta# discard
Changes have been discarded
vyatta@vyatta# show interfaces ethernet eth2
```

```
address 192.168.1.100/24
hw-id: 00:13:46:e6:f6:87
```

edit

Navigates to a subnode in the configuration tree for editing.

Syntax

edit *path*

Command Mode

Configuration mode.

Configuration Statement

None.

Parameters

path	The path to the node of configuration tree you want to edit.

Default

None.

Usage Guidelines

Use this command to navigate to a specific configuration subnode for editing. The [**edit**] prompt changes dynamically to mark your place in the configuration tree.

Once at that location, any actions you take such as showing, creating, or deleting configuration are relative to your location in the tree.

You can only navigate to a configuration node that has already been created and committed. Configuration nodes are created and modified using set command and are committed using commit command.

Examples

The following example begins at the top of the configuration tree in configuration mode and navigates to the **system login** configuration node. Once at the **system login** node, a **show** command displays just the contents of the **login** node.

In this example, notice how the prompt changes to mark the location in the configuration tree.

Example 2-14 Navigating in the configuration tree

```
vyatta@vyatta# edit system login
[edit system login]
vyatta@vyatta# show
user mike {
    authentication {
        encrypted-password $1$hccJixQo$V6sL5hDl6CUmVZvaH1vTf0
        plaintext-password ""
    }
}
user vyatta {
    authentication {
        encrypted-password $1$$Ht7gBYnxI1xCdO/JOnodh.
    }
}
[edit system login]
```

exit (configuration)

Navigates up one level of use.

- From a configuration subnode, jumps to the top of the configuration tree.
- From the top of the configuration tree, exits to operational mode.
- From operational mode, exits the system.

Syntax

exit [**discard**]

Command Mode

Configuration mode.

Operational mode.

Configuration Statement

None.

Parameters

discard	Applies when exiting from configuration mode to operational mode with uncommitted configuration changes. Allows you to exit from configuration mode by discarding all configuration changes.

Default

None.

Usage Guidelines

Use this command from a subnode in the configuration tree to navigate to the top of the configuration tree.

Use this command from the top of the configuration tree to exit from configuration mode to operational mode.

If you try to exit from configuration mode while there are still uncommitted configuration changes, the system will give you a warning. You will not be able to exit from configuration mode until you either commit the changes by issuing the **commit** statement, or you discard the changes using the **exit** command with the **discard** option. This is the only case where this option applies.

Use this command in operational mode to exit from the system.

load

Loads a saved configuration.

Syntax

load *file-name*

Command Mode

Configuration mode.

Configuration Statement

None.

Parameters

file-name	The name of the configuration file, including the full path to its location.

Default

None.

Usage Guidelines

Use this command to manually load a configuration previously saved to a file.

The loaded configuration becomes the working configuration and must be **commit**ted before it becomes the active configuration.

Configuration can be loaded from a hard disk (including a Flash disk or USB device), a TFTP server, an FTP server, an SCP server, or an HTTP server. Note that you cannot load an empty configuration file; the configuration file must contain at least one configuration node. Also, an error will be reported if an invalid configuration file is loaded.

The default configuration directory is **/config**.

The following table shows the syntax for file specification for different file locations.

Table 2-4 Specifying locations for the configuration file

Location	Specification
An absolute path	Use standard UNIX file specification.
A relative path	Specify the path name relative to the default configuration directory.
FTP server	Use the following syntax for *file-name*: ftp://*user:passwd@host/config-file* where *user* is the user name on the host, *passwd* is the password associated with the user name, *host* is the host name or IP address of the FTP server, and *config-file* is the configuration file, including the path. If you do not specify *user* and *passwd* you will be prompted for them.
SCP server	Use the following syntax for *file-name*: scp://*user:passwd@host/config-file* where *user* is the user name on the host, *passwd* is the password associated with the user name, *host* is the host name or IP address of the SCP server, and *config-file* is the configuration file, including the path. If you do not specify *user* and *passwd* you will be prompted for them.
HTTP server	Use the following syntax for *file-name*: http://*host/config-file* where *host* is the host name or IP address of the HTTP server, and *config-file* is the configuration file, including the path.
TFTP server	Use the following syntax for *file-name*: tftp://*host/config-file* where *host* is the host name or IP address of the TFTP server, and *config-file* is the configuration file, including the path relative to the TFTP root directory.

Examples

Example 2-15 loads the configuration file **testconfig** from the default configuration directory.

Example 2-15 Loading saved configuration from a file

```
vyatta@vyatta# load testconfig
Loading config file /config/testconfig...

Load complete.  Use 'commit' to make changes active.
[edit]
vyatta@vyatta#
```

merge

Merges a saved configuration with the active (running) configuration.

Syntax

merge *file-name*

Command Mode

Configuration mode.

Configuration Statement

None.

Parameters

file-name	The name of the configuration file, including the full path to its location.

Default

None.

Usage Guidelines

Use this command to manually load a configuration previously saved to a file and merge it with the active (running) configuration. The process of merging adds new configuration entries and applies any modifications to existing active entries to produce a new working configuration. This configuration must be **commit**ted before it becomes the active configuration.

Configuration can be loaded from a hard disk (including a Flash disk or USB device), a TFTP server, an FTP server, an SCP server, or an HTTP server. Note that you cannot load an empty configuration file; the configuration file must contain at least one configuration node.

The default configuration directory is **/config**.

The following table shows the syntax for file specification for different file locations.

Table 2-5 Specifying locations for the configuration file

Location	**Specification**
An absolute path	Use standard UNIX file specification.
A relative path	Specify the path name relative to the default configuration directory.
FTP server	Use the following syntax for *file-name*: ftp://*user:passwd@host/config-file* where *user* is the user name on the host, *passwd* is the password associated with the user name, *host* is the host name or IP address of the FTP server, and *config-file* is the configuration file, including the path. If you do not specify *user* and *passwd* you will be prompted for them.
SCP server	Use the following syntax for *file-name*: scp://*user:passwd@host/config-file* where *user* is the user name on the host, *passwd* is the password associated with the user name, *host* is the host name or IP address of the SCP server, and *config-file* is the configuration file, including the path. If you do not specify *user* and *passwd* you will be prompted for them.
HTTP server	Use the following syntax for *file-name*: http://*host/config-file* where *host* is the host name or IP address of the HTTP server, and *config-file* is the configuration file, including the path.
TFTP server	Use the following syntax for *file-name*: tftp://*host/config-file* where *host* is the host name or IP address of the TFTP server, and *config-file* is the configuration file, including the path relative to the TFTP root directory.

Examples

Example 2-16 loads the configuration file **testconfig** from the default configuration directory and merges it with the active configuration.

This new working configuration must be committed before it becomes active. After the merge, you must save the new file if you want to be able to load it again. If you want the system to load the merged configuration when it boots, you must save the file to **/config/config.boot**.

Example 2-16 Merging configuration from a file

```
vyatta@vyatta# merge testconfig
Loading config file /config/testconfig...

Merge complete.  Use 'commit' to make changes active.
[edit]
vyatta@vyatta#
```

rename

Allows you to change the identifier of a named configuration node.

Syntax

rename *from-config-node to to-config-node*

Command Mode

Configuration mode.

Configuration Statement

None.

Parameters

from-config-node	The configuration node to be renamed. The format is a series of space-separated tokens representing the path through the configuration hieratchy to the node; for example **firewall name RULE-SET-1 rule 10**.
to-config-node	The new identifier for the configuration node. The format is a series of space-separated tokens representing the path through the configuration hieratchy y to the new node; for example, **firewall name RULE-SET-1 rule 11**.

Default

None.

Usage Guidelines

Use this command in conjunction to rename (that is, to change the identifier of) a configuration node, such as a firewall rule set.

To make specifying the configuration subnode easier, use this command in conjunction with the **edit** command. Use the **edit** command to navigate to the appropriate place in the configuration hierarchy, then the appropriate subnode.

To make renaming the configuration node easier, use this command with the **edit** command. To do this, you move to the appropriate point in the configuration and then rename the appropriate subnode.

If you show configuration before it is committed, you will see the original configuration flagged with a minus sign (“-”) and the new configuration flagged with a plus sign (“+”); the flags and the original configuration node disappears after the configuration change is committed.

Examples

Example 2-17 renames rule 10 in firewall rule set RULE-SET-1 to rule 12.

Example 2-17 Renaming a configuration node

```
vyatta@vyatta# show firewall
 name RULE-SET-1 {
   rule 10 {
       action accept
   }
 }
vyatta@vyatta# edit firewall name RULE-SET-1
[edit firewall name RULE-SET-1]
vyatta@vyatta# rename rule 10 to rule 12
[edit firewall name RULE-SET-1]
vyatta@vyatta# show
-rule 10 {
-  action accept
-}
+rule 12 {
+  action accept
+}
[edit firewall name RULE-SET-1]
vyatta@vyatta# commit
[edit firewall name RULE-SET-1]
vyatta@vyatta# show
 rule 12 {
   action accept
 }
[edit firewall name RULE-SET-1]
vyatta@vyatta# top
```

rollback <rev-num>

Allows you to roll back configuration to a specific revision.

Syntax

rollback *rev-num*

Command Mode

Configuration mode.

Configuration Statement

None.

Parameters

rev-num	The configuration revision to roll back to.

Default

None.

Usage Guidelines

Use this command to roll back to the configuration revision specified.

NOTE *For the roll back to take effect, the system must be rebooted after the configuration is rolled back. A prompt will ask whether or not to reboot the system once the command completes.*

You can see the list of configuration file revisions using the show system commit operational mode command (use “run show system commit” from configuration mode).

save

Saves the running configuration to a file.

Syntax

save *file-name*

Command Mode

Configuration mode.

Configuration Statement

None.

Parameters

file-name	The name of the file where the information is to be saved, including the path to the file.

Default

None.

Usage Guidelines

Use this command to save the running configuration to a file.

The resulting file can later be loaded into the running system to replace the previous running configuration, using the load command. A non-absolute path is interpreted relative to the default configuration directory, which is **/config**.

The following table shows the syntax for file specification for different file locations.

Table 2-6 Specifying locations for the configuration file

Location	Specification
An absolute path	Use standard UNIX file specification.
A relative path	Specify the path name relative to the default configuration directory.

Table 2-6 Specifying locations for the configuration file

Location	Specification
FTP server	Use the following syntax for *file-name*: ftp://*user:passwd@host/config-file* where *user* is the user name on the host, *passwd* is the password associated with the user name, *host* is the host name or IP address of the FTP server, and *config-file* is the configuration file, including the path. If you do not specify *user* and *passwd* you will be prompted for them.
SCP server	Use the following syntax for *file-name*: scp://*user:passwd@host/config-file* where *user* is the user name on the host, *passwd* is the password associated with the user name, *host* is the host name or IP address of the SCP server, and *config-file* is the configuration file, including the path. If you do not specify *user* and *passwd* you will be prompted for them.
TFTP server	Use the following syntax for *file-name*: tftp://*host/config-file* where *host* is the host name or IP address of the TFTP server, and *config-file* is the configuration file, including the path relative to the TFTP root directory.

If you overwrite a configuration file, the system retains one backup, using a *file-name~* convention. For example, if you save over **my-config.boot**, the system moves the previous file to **my-config.boot~**.

Note that the **save** command only writes committed changes. If you makes configuration changes and try to save, the system warns you that you have uncommitted changes and then saves only the committed changes.

Examples

Example 2-18 saves the running configuration into the file **my-config** in the default configuration directory, exits from configuration mode, and displays the set of files stored in the configuration directory.

Example 2-18 Saving configuration to a file

```
vyatta@vyatta# save
Saving configuration to '/config/config.boot'...
Done
```

```
vyatta@vyatta# exit
vyatta@vyatta:~$ show files /config
total 24K
-rw-rw-r-- 1 vyatta xorp 2.8K Nov 28 10:30 config.boot
-rw-rw-r-- 1 vyatta xorp 2.8K Nov 27 14:32 config.boot~
-rw-rw-r-- 1 vyatta xorp 2.8K Nov 28 10:30 my-config
-rw-rw-r-- 1 vyatta xorp 2.8K Nov 27 21:50 my-config~
vyatta@vyatta:~$
```

Example 2-19 saves the current running configuration to the file **my-config** in the root directory of a TFTP server at 10.1.0.35.

Example 2-19 Saving configuration to a file on a TFTP server

```
vyatta@vyatta# save tftp://10.1.0.35/my-config
Saving configuration to 'tftp://10.1.0.35/my-config'...
Done
vyatta@vyatta#
```

set

Creates a new configuration node, or modifies a value in an existing configuration node.

Syntax

To create a new configuration node, the syntax is as follows:

set *config-node* [*identifier*]

To set an attribute within a configuration node, the syntax is as follows:

set *config-node* [*identifier*] *attribute* [*value*]

Command Mode

Configuration mode.

Configuration Statement

None.

Parameters

config-node	The configuration node to be created or modified, including the full path, separated by spaces, through the configuration hierarchy to the node.
identifier	The identifier of the configuration node. Mandatory if the configuration node has an identifier; forbidden otherwise.
attribute	The configuration attribute or property to be set. If the attribute statement does not exist, it is created. If the attribute statement already exists, the value is set to the new value.
value	The new value of the attribute. Mandatory if the attribute statement requires a value; forbidden otherwise.

Default

None.

Usage Guidelines

Use this command to add a configuration element to the current configuration—for example, to enable a routing protocol or define an interface.

You can also use this command to modify the value of an existing configuration item. When setting configuration values, note that the change does not take effect until the change is committed, using commit command.

Once a configuration node has been added, you can modify it later using set command, or delete it using delete command.

Examples

Example 2-20 adds a configuration node for an Ethernet interface and commits the change.

Example 2-20 Adding a configuration node

```
vyatta@vyatta# set interfaces ethernet eth1 address 192.150.187.108/24
vyatta@vyatta# commit
```

show

Displays configuration information in configuration mode.

Syntax

show [-**all**] *config-node*

Command Mode

Configuration mode.

Configuration Statement

None.

Parameters

config-node	The configuration node you want to view, including the path. The node must exist and the created node must have been committed. The configuration node specification is interpreted relative to your current position in the configuration tree.
–all	Includes default information in the displayed information.

Default

When used with no configuration node specification, this command displays all existing configuration nodes and sub-nodes starting from your current location in the configuration tree.

When used without the **–all** option, default information is not shown

Usage Guidelines

Use this command in configuration mode to display the configured state of the system.

This command displays the specified configuration node and all sub-nodes. The node specification is interpreted relative to your current location in the configuration tree.

Unless the **–all** keyword is used, default information is not included in displayed information.

In addition to this command, there are a number of **show** commands available in operational mode. For a list of these commands, please see the Quick List of Commands, which begins on page xiv.

Examples

Example 2-21 shows the **service** node displayed using the **show** command in configuration mode.

Example 2-21 Displaying configuration information

```
vyatta@vyatta# show service
dhcp-server {
}
dns {
}
ssh {
}
telnet {
}
vyatta@vyatta#
```

show configuration (operational)

Displays system configuration from operational mode.

Syntax

show configuration [**all** | **commands** | **files**]

Command Mode

Operational mode.

Parameters

all	Displays all configuration, including default values that would not normally be displayed.
commands	Displays the running configuration as a list of **set** commands. These are the commands that would be used to generate the configuration from scratch.
files	Displays a list of configuration files in **/config**.

Default

Displays only the values that have been set explicitly (that is, non-default values).

Usage Guidelines

Use this command to list configuration information while remaining in operational mode.

Using **show configuration** in operational mode is equivalent to using **show** in configuration mode.

Examples

Example 2-22 displays the configuration from operational mode. (For brevity, only the first screen of the information is shown.)

Example 2-22 Displaying configuration information in operational mode

```
vyatta@vyatta:~$ show configuration
interfaces {
    ethernet eth0 {
        address 192.168.1.77/24
        hw-id 00:0c:29:68:b3:9f
    }
    ethernet eth1 {
        hw-id 00:0c:29:68:b3:a9
    }
    loopback lo {
    }
}
service {
    ssh {
    }
}
system {
    gateway-address 192.168.1.254
    host-name R1
    login {
        user vyatta {
            authentication {
                encrypted-password ****************
:
```

show system commit

Displays a summary of configuration file revisions.

Syntax

show system commit

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display a summary of configuration file revisions.

Examples

Example 2-23 shows the commit history of system R1.

Example 2-23 Displaying the commit history

```
vyatta@R1:~$ show system commit
0   2010-11-15 16:55:17 by vyatta via cli
    delete firewall
1   2010-11-15 16:54:40 by vyatta via cli
2   2010-11-15 16:54:40 by root via cli
    baseline
vyatta@R1:~$
```

show system commit diff <rev-num>

Compares adjacent configuration file revisions.

Syntax

show system commit diff *rev-num*

Command Mode

Operational mode.

Parameters

rev-num	The configuration file revision to compare with a subsequent revision; that is: *rev-num* +1.

Default

None.

Usage Guidelines

Use this command to compare two revisions of the configuration file.

The revisions to be compared are *rev-num* and *rev-num*+1. This is a shortcut for the command **show system commit file** *rev_num* **compare** *rev-num*+1. You can see the list of configuration file revisions using the show system commit command.

Examples

Example 2-24 shows two configuration file revisions (3 and 4) being compared on R1.

Example 2-24 Displaying the comparison of two configuration file revisions

```
vyatta@R1:~$ show system commit diff 3
@@ -84,6 +84,9 @@
       }
   }
   system {
+      config-management {
+          commit-revisions 20
+      }
       console {
           device ttyS0 {
               speed 9600
vyatta@R1:~$
```

show system commit file <rev-num>

Displays a specific configuration file revision.

Syntax

show system commit file *rev-num* [**compare** *rev-num1*]

Command Mode

Operational mode.

Parameters

rev-num	The configuration file revision number to display.
rev-num1	The configuration file revision number to compare with.

Default

None.

Usage Guidelines

Use this command to display a configuration file revision. Use the **compare** option to compare two revisions of the configuration file. You can see the list of configuration file revisions using the show system commit command.

Examples

Example 2-25 shows revision 0 of the configuration file on R1.

Example 2-25 Displaying revision 0 of the configuration file

```
vyatta@R1:~$ show system commit file 0
  interfaces {
      ethernet eth0 {
          address dhcp
          description "bridge to io"
          duplex auto
          smp_affinity auto
          speed auto
      }
```

```
[... the rest of the configuration file]
vyatta@R1:~$
```

Example 2-26 shows two configuration file revisions (3 and 4) being compared on R1.

Example 2-26 Displaying the comparison of two configuration file revisions

```
vyatta@R1:~$ show system commit file 3 compare 4
@@ -84,6 +84,9 @@
       }
   }
   system {
+      config-management {
+          commit-revisions 20
+      }
       console {
           device ttyS0 {
               speed 9600
vyatta@R1:~$
```

system config-management commit-archive location <location>

Enables archiving of configuration to the specified location every time a change is committed.

Syntax

set system config-management commit-archive location *location*

delete system config-management commit-archive location *location*

show system config-management commit-archive location

Command Mode

Configuration mode.

Configuration Statement

```
system {
    config-management {
        commit-archive {
            location location
        }
    }
}
```

Parameters

location	Multi-node. Location of the configuration archive. Supported file transfer methods and their general formats are as follows: **scp**://*user*:*passwd*@*host*/*dir* **ftp**://*user*:*passwd*@*host*/*dir* **tftp**://*host*/*dir* where *user* is the user name on the host, *passwd* is the password associated with the user name, *host* is the host name or IP address of the remote server, and *dir* is the directory path to save the file in. The saved file will contain the original file name (config.boot) followed by the hostname of the local system, the date (YYYYMMDD), and the time (HHMMSS). For example: config.boot-R1.20110126_193402 is the config.boot file from R1 saved on Jan 26, 2011 at 7:34:02pm. You can define more than one archive location by creating multiple **location** configuration nodes.

Default

When this option is not set, system configuration is not archived on commit.

Usage Guidelines

Use this command to enable automatic remote archiving of configuration on commit.

The system automatically archives configuration on commit. These archives are stored locally in the /config/archive directory and the number of revisions to keep is set using the system config-management commit-revisions <revisions> command.

The system config-management commit-archive location <location> command allows you to archive an unlimited number of configuration revisions to a remote location using either FTP, SCP, or TFTP as the file transfer method. The archive operation occurs in the forground.

Use the **set** form of this command to enable remote configuration revision archiving and specify the location of the archive.

Use the **delete** form of this command to disable remote configuration revision archiving.

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

system config-management commit-revisions <revisions>

Specifies the number of configuration revisions to store locally.

Syntax

set system config-management commit-revisions *revisions*

delete system config-management commit-revisions

show system config-management commit-revisions

Command Mode

Configuration mode.

Configuration Statement

```
system {
   config-management {
      commit-revisions revisions
   }
}
```

Parameters

revisions	The maximum number of configuration revisions to store locally. The default is 20.

Default

By default, 20 configuration revisions are stored.

Usage Guidelines

Use this command to specify the maximum number of configuration revisions to store locally.

The system automatically stores revisions of system configuration every time a configuration change is committed. These revisions are stored in the **/config/archive** directory. This command sets the number of revisions to be stored.

A new revision is stored each time the configuration is committed. Once the maximum number of revisions has been reached, the oldest revision is removed to make way for new revisions.

Note that you can store an unlimited number of configuration revisions to a remote location using the system config-management commit-archive location <location> command.

Use the **set** form of this command to change the number or locally stored configuration revisions.

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

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

top

Moves to the top level of the configuration hierarchy.

Syntax

top

Command Mode

Configuration mode.

Configuration Statement

None.

Parameters

None.

Usage Guidelines

Use this command to quickly navigate to the top level of configuration mode.

Examples

Example 2-27 navigates down through several nodes of the configuration tree, then uses the **top** command to jump directly to the top of the tree. In this example, notice how the [**edit**] line displays your location in the configuration tree.

Example 2-27 Navigating to the top of the configuration tree

```
vyatta@vyatta# edit protocols rip interface eth0
[edit protocols/rip/interface/eth0]
vyatta@vyatta# top
vyatta@vyatta#
```

up

Navigates up one level in the configuration tree.

Syntax

up

Command Mode

Configuration mode.

Configuration Statement

None.

Parameters

None.

Usage Guidelines

Use this command to navigate one level up in configuration mode.

Examples

Example 2-28 navigates down through several nodes of the configuration tree, then uses the **up** command to navigate successively higher in the tree. In this example, notice how the [edit] line displays your location in the configuration tree.

Example 2-28 Navigating up a level in the configuration tree

```
vyatta@vyatta# edit protocols rip interface eth0
[edit protocols/rip/interface/eth0]
vyatta@vyatta# up
[edit protocols/rip/interface]
vyatta@vyatta# up
[edit protocols/rip/]
```

Chapter 3: System Management

This chapter describes Vyatta system features for basic system management tasks, such as setting host information, working with the ARP cache, and setting the system date and time.

This section presents the following topics:

- Basic System Configuration
- Monitoring System Information
- System Management Commands

Basic System Configuration

The commands in this chapter allow you to change and view basic IP system information. This section presents the following topics:

- Configuring Host Information
- Configuring DNS
- Configuring Date and Time
- Monitoring System Information

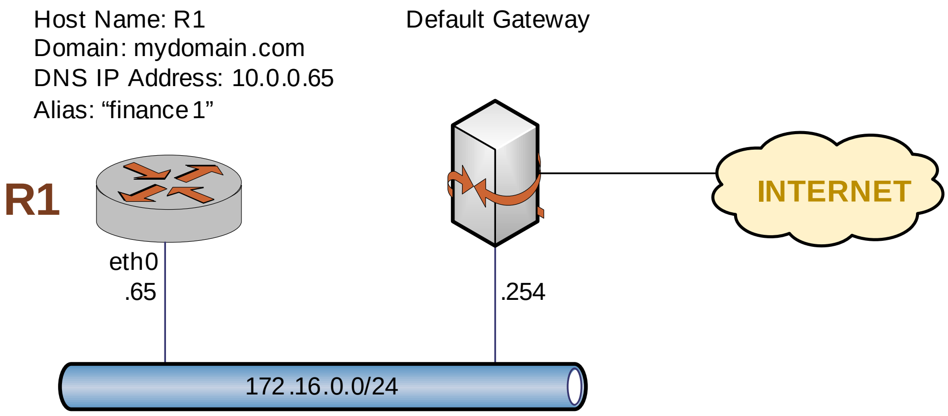
Configuring Host Information

This section presents the following topics:

- Host Name
- Domain
- IP Address
- Default Gateway
- Aliases

In this section, sample configurations are presented for the system's host information. The sample configuration used is shown in Figure 3-1.

Figure 3-1 Host information



This section includes the following examples:

- Example 3-1 Setting the system's host name
- Example 3-2 Setting the system's domain

- Example 3-3 Mapping the system's IP address to its host name
- Example 3-4 Setting the default gateway
- Example 3-5 Creating an alias for the system

Host Name

The Vyatta System's name is set using the **system host-name** command. System names can include letters, numbers, and hyphens ("-").

Example 3-1 sets the system's host name to R1. To set the system host name, perform the following steps in configuration mode:

Example 3-1 Setting the system's host name

Step	Command
Set the system's host name.	`vyatta@vyatta# set system host-name R1`

Commit the change. The command prompt changes to reflect the change	`vyatta@vyatta# commit`
	`vyatta@R1#`

Show the configuration.	`vyatta@R1# show system host-name`
	`  host-name R1`

Domain

The system's domain is set using the **system domain-name** command. Domain names can include letters, numbers, hyphens, and periods.

***NOTE** **system domain-name** and **system domain-search** are mutually exclusive. Only one of the two can be configured at any one time.*

Example 3-2 sets the system's domain to **mydomain.com**.

To set the system's domain, perform the following steps in configuration mode:

Example 3-2 Setting the system's domain

Step	Command
Set the domain name.	`vyatta@R1# set system domain-name mydomain.com`
Commit the change.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show system domain-name`
	`  domain-name mydomain.com`

IP Address

The system's IP address can be statically mapped to its host name for local DNS purposes, using the **system static-host-mapping** command.

IP networks are specified in CIDR format—that is, in *ip-address/prefix* notation such as 192.168.12.0/24. For single addresses, use dotted quad format, that is, *a.b.c.d*. For network prefixes, enter a decimal number from 1 through 32.

A good practice is to map the system's host name to the loopback address, as the loopback interface is the most reliable on the system. In this example, the loopback interface is given the address 10.0.0.65. This is the address configured for the loopback interface in the sample topology used in this guide.

Example 3-3 creates a static mapping between the host name R1 and IP address 10.0.0.65. This is the IP address the DNS server will use to resolve DNS requests for **R1.mydomain.com**.

To map the host name to the IP address, perform the following steps in configuration mode:

Example 3-3 Mapping the system's IP address to its host name

Step	Command
Map host name R1 to IP address 10.0.0.65.	`vyatta@R1# set system static-host-mapping host-name R1 inet 10.0.0.65`
Commit the change.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show system static-host-mapping`
	`  host-name R1 {`
	`      inet 10.0.0.65`
	`  }`

Default Gateway

Example 3-4 specifies a default gateway for the system at 172.16.0.254.

To specify the default gateway, perform the following steps in configuration mode:

Example 3-4 Setting the default gateway

Step	Command
Specify the default gateway.	`vyatta@R1# set system gateway-address 172.16.0.254`
Commit the change.	`vyatta@R1# commit`

Example 3-4 Setting the default gateway

Step	Command
Show the configuration.	vyatta@R1# **show system gateway-address** gateway-address 172.16.0.254

Aliases

You can define one or more aliases for the system by mapping the system's IP address to more than one host name.

Example 3-5 creates the alias **finance1** for the system.

To create an alias for the system, perform the following steps in configuration mode:

Example 3-5 Creating an alias for the system

Step	Command
Define an alias.	vyatta@R1# **set system static-host-mapping host-name R1 alias finance1**
Commit the change.	vyatta@R1# **commit**

Show the configuration.	vyatta@R1# **show system static-host-mapping**
	host-name R1 {
	alias finance1
	inet 10.0.0.65
	}

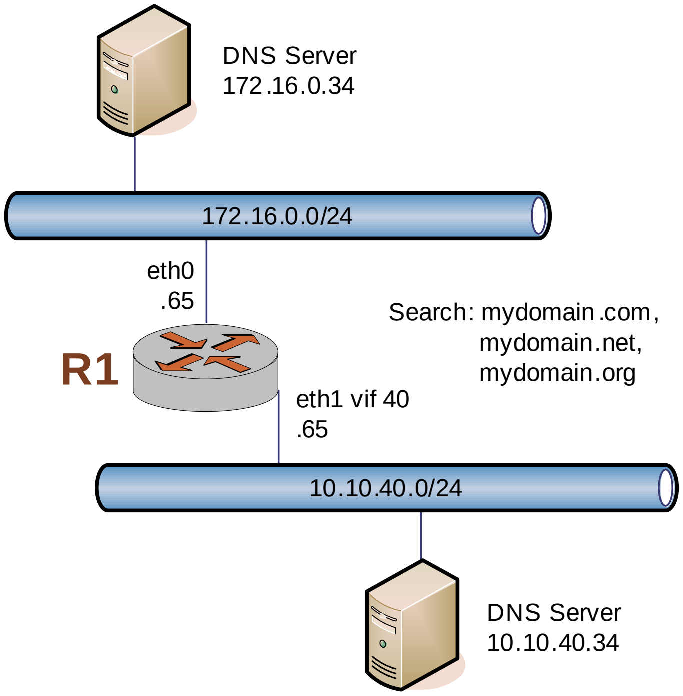
Configuring DNS

This section presents the following topics:

- DNS Name Servers
- Domain Search Order

In this section, sample configurations are presented for DNS information. The DNS configuration used is shown in Figure 3-2.

Figure 3-2 DNS



This section includes the following examples:

- Example 3-6 Specifying DNS name servers
- Example 3-7 Setting search order for domain completion

DNS Name Servers

DNS name servers are specified using the **system name-server** command.

NOTE *The order in which the DNS name servers are added to the configuration is the order in which they are accessed.*

Example 3-6 specifies two DNS servers for the system: one at 172.16.0.34, and the other at 10.10.40.34.

To specify DNS servers, perform the following steps in configuration mode:

Example 3-6 Specifying DNS name servers

Step	Command
Specify the first DNS server.	`vyatta@R1# set system name-server 172.16.0.34`

Example 3-6 Specifying DNS name servers

Step	Command
Specify the second DNS server.	vyatta@R1# **set system name-server 10.10.40.34**
Commit the change.	vyatta@R1# **commit**

Show configuration.	vyatta@R1# **show system name-server**
	  name-server 172.16.0.34
	  name-server 10.10.40.34

Domain Search Order

You can specify a list of domains for the system to use to complete an unqualified host name. To define this list, specify the order in which domains are searched using the **system domain-search** command.

***NOTE** **system domain-name** and **system domain-search** are mutually exclusive. Only one of the two can be configured at any one time.*

The **system domain-search** command requires you to enter each domain name separately, specified in the order you want them searched. A domain name can include letters, numbers, hyphens ("-"), and periods (".").

Example 3-7 directs the system to attempt domain completion in the following order: first, mydomain.com; second, mydomain.net; and last mydomain.org.

To specify domain search order, perform the following steps in configuration mode:

Example 3-7 Setting search order for domain completion

Step	Command
Specify the first domain name.	vyatta@R1# **set system domain-search domain mydomain.com**
Specify the second domain name.	vyatta@R1# **set system domain-search domain mydomain.net**
Specify the third domain name.	vyatta@R1# **set system domain-search domain mydomain.org**
Commit the change.	vyatta@R1# **commit**

Show the configuration.	vyatta@R1# **show system domain-search**
	  domain mydomain.com
	  domain mydomain.net
	  domain mydomain.org

Configuring Date and Time

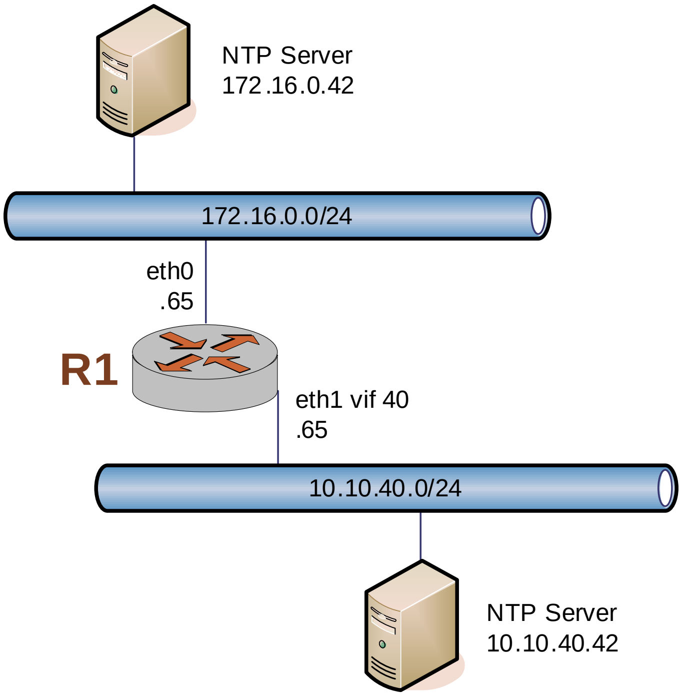
This section presents the following topics:

- Setting the Date
- Manually Synchronizing with an NTP Server
- Setting the Time Zone
- Using NTP for Automatic Synchronization

Date and time can either be set manually, or obtained by manually or automatically synchronizing the system with one or more Network Time Protocol (NTP) servers. Time zone must be manually set, and may be specified as an offset from Universal Coordinated Time (UTC) or as one of a number of supported literal time zones.

In this section, sample configurations are presented for maintaining date and time information. The sample configuration used is shown in Figure 3-3.

Figure 3-3 Date and time



This section includes the following examples:

- Example 3-8 Setting the date and time manually
- Example 3-9 Manually synchronizing the system with an NTP server
- Example 3-10 Setting the time zone as a Region/Location
- Example 3-11 Using NTP for automatic synchronization

Setting the Date

Example 3-8 manually sets the date to 1:15 PM exactly on April 24, 2007. The format is *MMDDhhmmCCYY*. Alternate formats are *MMDDhhmm*, *MMDDhhmmYY*, and *MMDDhhmmCCYY.ss*.

To manually set the date, perform the following steps in operational mode:

Example 3-8 Setting the date and time manually

Step	Command
Specify the date. The format is *MMDDhhmmCCYY*.	vyatta@R1:~$ **set date 042413152007** Tue Apr 24 13:15:00 GMT 2007 vyatta@R1:~$

Manually Synchronizing with an NTP Server

Example 3-9 manually synchronizes the system clock with the NTP server at 172.16.0.42.

Note that this merely performs a one-time synchronization. It does not set up an ongoing association with the NTP server. For information about setting up automatic synchronization, please see "Using NTP for Automatic Synchronization" on page 107.

To perform a one-time synchronization with an NTP server, perform the following steps in operational mode:

Example 3-9 Manually synchronizing the system with an NTP server

Step	Command
Specify the location of the NTP server.	vyatta@R1:~$ **set date ntp 172.16.0.42** Tue Apr 24 13:15:00 UTC 2007 vyatta@R1:~$

Setting the Time Zone

Time zone must be configured, using **system time-zone** command. To do this, you specify the Region/Location that best defines your location. For example, specifying **US/Pacific** sets the time zone to US Pacific time. Command completion (i.e. the <Tab> key) can be used to list available time zones. The adjustment for daylight time will take place automatically based on the time of year.

Example 3-10 sets the time zone to Pacific time.

To set the time zone, perform the following steps in configuration mode:

Example 3-10 Setting the time zone as a Region/Location

Step	Command

Set the time zone.	`vyatta@R1#` **`set system time-zone US/Pacific`**
	`vyatta@R1#`

Commit the information.	`vyatta@R1#` **`commit`**

Show the configuration.	`vyatta@R1#` **`show system time-zone`**
	`    time-zone US/Pacific`

Using NTP for Automatic Synchronization

To use NTP for automatic synchronization, you must create associations with the NTP servers. To create an association with an NTP server, use the **system ntp server** command and specify the IP address of the server.

Example 3-11 configures two NTP servers: one at 172.16.0.42, and one at 10.10.40.42.

To specify NTP servers, perform the following steps in configuration mode:

Example 3-11 Using NTP for automatic synchronization

Step	Command
Specify a server at 172.16.0.42.	`vyatta@R1#` **`set system ntp server 172.16.0.42`**
Specify a server at 10.10.40.42.	`vyatta@R1#` **`set system ntp server 10.10.40.42`**
Commit the information.	`vyatta@R1#` **`commit`**

Example 3-11 Using NTP for automatic synchronization

Step	Command

Show the configuration. (Output is abbreviated here.)	`vyatta@R1#` **`show system`**
	`    host-name R1`
	`    domain-search {`
	`        domain mydomain.com`
	`        domain mydomain.net`
	`        domain mydomain.org`
	`    }`
	`    name-server 172.16.0.34`
	`    name-server 10.10.40.34`
	`    time-zone US/Pacific`
	`    ntp {`
	`        server 172.16.0.42`
	`        server 10.10.40.42`
	`    }`

Monitoring System Information

This section presents the following topics:

- Showing Host Information
- Showing the Date and Time

This section includes the following examples:

- Example 3-12 Showing the system host name
- Example 3-13 Showing the system date and time

Showing Host Information

To view the configured host name, use the **show host name** command in operational mode, as shown in Example 3-12:

Example 3-12 Showing the system host name

```
vyatta@R1:~$ show host name
R1
vyatta@R1:~$
```

Showing the Date and Time

To view the time according to the system clock, use the **show host date** command in operational mode, as shown in Example 3-13:

Example 3-13 Showing the system date and time

```
vyatta@R1:~$ show host date
Tue Apr 24 22:23:07 GMT+8 2007
vyatta@R1:~$
```

System Management Commands

This section presents the following commands.

Configuration Commands	
system console device <device>	Defines the specified device as a system console.
system console powersave	Enables blank screen powersaving on VGA console screens.
system domain-name <domain>	Sets the system's domain.
system domain-search domain <domain>	Defines a set of domains for domain completion.
system gateway-address <address>	Specifies the default gateway for the system.
system host-name <name>	Sets the host name for the system.
system name-server <address>	Specifies the DNS name servers available to the system.
system ntp server <name>	Specifies the NTP servers to use when synchronizing the system's clock.
system options reboot-on-panic <value>	Allows you set system behavior on system panic.
system static-host-mapping host-name <name>	Defines a static mapping between a host name and an IP address.
system time-zone <zone>	Sets the time zone for the local system clock.
Operational Commands	
clear arp address <ipv4>	Clears the system's ARP cache for the specified IP address.
clear arp interface <ethx>	Clears the system's ARP cache for the specified interface.
clear console	Clears the user's console.
clear interfaces counters	Clears interface counters for all interfaces.
init-floppy	Formats a floppy diskette and prepares it to receive a configuration file.
reboot	Reboots the system.
set date	Sets the system date and time directly or specifies an NTP server to acquire it from.
show arp	Displays the system's ARP cache.
show date	Displays the system date and time.

show hardware cpu	Displays information about the system's processor.
show hardware dmi	Displays information about the system's DMI.
show hardware mem	Displays information about the system's memory.
show hardware pci	Displays information about the system's PCI bus.
show history	Displays command execution history.
show host	Displays host information for hosts reachable by the system.
show interfaces	Displays information about system interfaces.
show license	Displays Vyatta license information.
show ntp	Shows the status of configured NTP servers.
show reboot	Shows the next scheduled reboot date and time.
show system boot-messages	Displays boot messages generated by the kernel.
show system connections	Displays active network connections on the system.
show system kernel-messages	Displays messages in the kernel ring buffer.
show system memory	Displays system memory usage.
show system processes	Displays active system processes.
show system routing-daemons	Displays active routing daemons.
show system storage	Displays system file system usage and available storage space.
show system uptime	Displays information on how long the system has been running.
show system usb	Displays information about peripherals connected to the USB bus.
show tech-support	Provides a consolidated report of system information.
show version	Displays information about the version of system software.
terminal	Controls behaviors of the system terminal.

Some commands related to certain features of system management are located in other locations:

Related Commands Documented Elsewhere	
system login	User management commands are described in Chapter 4: User Management.
system syslog	System logging commands are described in Chapter 7: Logging.

clear arp address <ipv4>

Clears the system's ARP cache for the specified IP address.

Syntax

clear arp address *ipv4*

Command Mode

Operational mode.

Parameters

ipv4	Removes the ARP entry for the specified IP address from the ARP cache.

Default

None.

Usage Guidelines

Use this command to remove ARP entries associated with a specific IP address from the ARP cache.

clear arp interface <ethx>

Clears the system's ARP cache for the specified interface.

Syntax

clear arp interface *eth0..eth23*

Command Mode

Operational mode.

Parameters

eth0..eth23	Clears the entire ARP cache for the specified Ethernet interface. The range of values is **eth0** to **eth23**.

Default

None.

Usage Guidelines

Use this command to remove ARP entries associated with an Ethernet interface from the ARP cache.

clear console

Clears the user’s console.

Syntax

clear console

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to clear the screen of the console.

clear interfaces counters

Clears interface counters for all interfaces.

Syntax

clear interfaces counters

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to clear the counters for all interfaces of all types, including ADSL, bridge, Ethernet, loopback, multilink, serial, and tunnel.

init-floppy

Formats a floppy diskette and prepares it to receive a configuration file.

Syntax

init-floppy

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to format a disk in the floppy disk drive.

The system puts a file system on the floppy disk and makes it accessible to the Vyatta system. It also saves a copy of the running configuration to **/media/floppy/config/config.boot**.

Initializing the floppy disk erases any previous data on the disk. The system reminds you of this, and provides a 5-second window in which you can quit out of the command by typing "y" in response to the question "Continue (y/n)? [y]" or pressing <Ctrl>+c.

Once the floppy disk has been formatted, the **config.boot** file is automatically saved to it. You can also save the **config.boot** configuration file to disk using the **save** command.

Examples

Example 3-14 prepares a floppy disk for receiving a configuration file and saves the running configuration to **/media/floppy/config/config.boot**.

Example 3-14 Initializing a floppy diskette for saving configuration files

```
vyatta@R1:~$ init-floppy
This will erase all data on floppy /dev/fd0.
```

```
Your configuration was saved in: /media/floppy/config/config.boot
vyatta@R1:~$
```

reboot

Reboots the system.

Syntax

reboot [**at** *time* | **cancel**]

Command Mode

Operational mode.

Parameters

at *time*	The time the system is scheduled to reboot. Set the date and/or time directly using one of the following formats: hh:mm MMDDYY “hh:mm MMDDYY” **midnight** **noon** “**now +** x units” Note that the hour field (hh) uses the 24 hour clock (e.g. 3:00 pm would be represented as 15 in the hour field). Note that units can be **minutes**, **hours**, **days**, **weeks**, **months**, or **years**.
cancel	Cancels a previously scheduled reboot.

Default

None.

Usage Guidelines

Use this command to reboot the system.

Before the system reboots, a message is broadcast to all logged on users warning them of the reboot.

Only users with admin level permissions can execute this command.

Examples

Example 3-15 reboots the system.

Example 3-15 Rebooting the system

```
vyatta@R1:~$ reboot
Proceed with reboot? [confirm]y

Broadcast message from root@R1 (tty1) (Mon Jan 21 17:52:37 2008):

The system is going down for reboot NOW!
```

Example 3-16 reboots the system at the current time on a specified date.

Example 3-16 Rebooting the system at a specified date

```
vyatta@R1:~$ reboot at 121109

Reload scheduled for at Saturday Dec 12 20:18:00 2009

Proceed with reboot schedule? [confirm]y

Reload scheduled for at Saturday Dec 12 20:18:00 2009
```

Example 3-17 cancels a scheduled reboot.

Example 3-17 Cancel a scheduled reboot

```
vyatta@R1:~$ reboot cancel
Reboot canceled
vyatta@R1:~$
```

set date

Sets the system date and time directly or specifies an NTP server to acquire it from.

Syntax

set date {*datetime* | **ntp** *ntpserver*}

Command Mode

Operational mode.

Parameters

datetime	Set the date and time directly using one of the following formats: MMDDhhmm MMDDhhmmYY MMDDhhmmCCYY MMDDhhmmCCYY.ss Note that the hour field (hh) uses the 24 hour clock (e.g. 3:00 pm would be represented as 15 in the hour field).
ntpserver	Specifies a Network Time Protocol (NTP) to acquire the current time from. You can specify either an IPv4 address or a hostname to identify the NTP server.

Default

None.

Usage Guidelines

Use this command to set the system date and time either directly or by specifying a Network Time Protocol (NTP) server to acquire the date and time from. If a timezone has not been configured then GMT is assumed. The timezone is set using the system time-zone <zone> command (see page 184).

Examples

Example 3-18 sets the system date and time to May 15, 2008 at 10:55 pm (assuming that the timezone is set to Pacific Daylight Time).

Example 3-18 Set the date and time directly

```
vyatta@R1:~$ set date 051522552008
Thu May 15 22:55:00 PDT 2008
vyatta@R1:~$
```

Example 3-19 sets the system date and time using an NTP server.

Example 3-19 Set the date and time using an NTP server

```
vyatta@R1:~$ set date ntp 69.59.150.135
15 May 23:00:00 ntpdate[7038]: step time server 69.59.150.135 offset
425.819267 sec
vyatta@R1:~$
```

show arp

Displays the system’s ARP cache.

Syntax

show arp [*interface*]

Command Mode

Operational mode.

Parameters

interface	Show ARP information for the specified interface.

Default

None.

Usage Guidelines

Use this command to display the system’s ARP cache.

Table 3-1 shows possible ARP states.

Table 3-1 ARP states

State	Description
incomplete	Address resolution is currently being preformed on this neighbor entry.
reachable	Indicates that the neighbor is reachable. Positive confirmation has been received and the path to this neighbor is operational.
stale	More than the configured elapsed time has passed since reachability confirmation was received from this neighbor.
delay	More than the configured elapsed time has passed since reachability confirmation was received from this neighbor. This state allows TCP to confirm the neighbor. If not, a probe should be sent after the next delay time has elapsed.
probe	A solicitation has been sent and the system is waiting for a response from this neighbor.

Table 3-1 ARP states

State	Description
failed	Neighbor reachability state detection failed.
noarp	This is a pseudo-state, indicating that ARP is not used for this neighbor entry.
permanent	This is a pseudo-state indicating that this entry should not be cleared from the cache.
none	No state is defined.

Examples

Example 3-20 shows the ARP cache of systemR1.

Example 3-20 Displaying the ARP cache

```
vyatta@R1:~$ show arp
Address         HWtype    HWaddress            Flags  MaskIface
172.16.215.1    ether     00:12:D9:74:BE:91    C      eth0
10.1.0.1        ether     00:04:23:09:0F:79    C      eth0
vyatta@R1:~$
```

show date

Displays the system date and time.

Syntax

show date [**utc**]

Command Mode

Operational mode.

Parameters

utc	Shows the date and time in Coordinated Universal Time.

Default

None.

Usage Guidelines

Use this command to display the system date and time in either local time or UTC time.

Examples

Example 3-21 shows the system date and time on R1.

Example 3-21 Displaying the system date and time

```
vyatta@R1:~$ show date
Tue May 20 17:27:07 PDT 2008
vyatta@R1:~$
```

show hardware cpu

Displays information about the system's processor.

Syntax

show hardware cpu [**summary**]

Command Mode

Operational mode.

Parameters

summary	Show the CPUs on the system.

Default

None.

Usage Guidelines

Use this command to view information about the processor(s) used in the system's hardware platform.

Examples

Example 3-22 shows CPU information on R1.

Example 3-22 Showing CPU information

```
vyatta@R1:~$ show hardware cpu
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 15
model name      : Intel(R) Xeon(R) CPU           E5310  @ 1.60GHz
stepping        : 8
cpu MHz         : 1595.101
cache size      : 4096 KB
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
```

```
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 10
wp              : yes
flags          : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss nx constant_tsc up
arch_perfmon pebs bts pni ds_cpl ssse3 dca
bogomips        : 3213.51
clflush size    : 64
power management:

vyatta@R1:~$
```

show hardware dmi

Displays information about the system's DMI.

Syntax

show hardware dmi

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to view information about the system's desktop management interface (DMI). The DMI provides a standard framework for managing resources in the device.

Examples

Example 3-23 shows DMI information on R1.

Example 3-23 Showing DMI information

```
vyatta@R1:~$ show hardware dmi
bios_date: 04/17/2006
bios_vendor: Phoenix Technologies LTD
bios_version: 6.00
board_asset_tag:
board_name: 440BX Desktop Reference Platform
board_vendor: Intel Corporation
board_version: None
chassis_asset_tag: No Asset Tag
chassis_type: 1
chassis_vendor: No Enclosure
chassis_version: N/A
```

```
product_name: VMware Virtual Platform
product_version: None
sys_vendor: VMware, Inc.
vyatta@R1:~$
```

show hardware mem

Displays information about the system's memory.

Syntax

```
show hardware mem
```

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to view information about the system memory.

Examples

Example 3-24 shows memory information on R1.

Example 3-24 Showing memory information

```
vyatta@R1:~$ show hardware mem
MemTotal:        515972 kB
MemFree:         341468 kB
Buffers:          28772 kB
Cached:          116712 kB
SwapCached:           0 kB
Active:           35912 kB
Inactive:        117272 kB
HighTotal:            0 kB
HighFree:             0 kB
LowTotal:        515972 kB
LowFree:         341468 kB
SwapTotal:            0 kB
SwapFree:             0 kB
```

```
Dirty:               0 kB
Writeback:           0 kB
AnonPages:        7700 kB
Mapped:           4048 kB
Slab:            14644 kB
SReclaimable:     9440 kB
SUnreclaim:       5204 kB
PageTables:        288 kB
NFS_Unstable:        0 kB
Bounce:              0 kB
CommitLimit:    257984 kB
Committed_AS:    21636 kB
VmallocTotal:   507896 kB
VmallocUsed:      3896 kB
VmallocChunk:   503932 kB
vyatta@R1:~$
```

show hardware pci

Displays information about the system's PCI bus.

Syntax

show hardware pci [detailed]

Command Mode

Operational mode.

Parameters

detailed	Shows detailed information about the PCI bus.

Default

None.

Usage Guidelines

Use this command to view information about the peripheral component interconnect (PCI) bus. The PCI provides communication among the system's peripheral components and the processor.

Examples

Example 3-25 shows PCI information on R1.

Example 3-25 Showing PCI bus information

```
vyatta@R1:~$ show hardware pci
00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host
bridge (rev 01)
00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX AGP
bridge (rev 01)
00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 08)
00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:0f.0 VGA compatible controller: VMware Inc Abstract SVGA II Adapter
00:10.0 SCSI storage controller: LSI Logic / Symbios Logic 53c1030 PCI-X
Fusion-MPT Dual Ultra320 SCSI (rev 01)
```

```
00:11.0 Ethernet controller: Advanced Micro Devices [AMD] 79c970 [PCnet32
LANCE] (rev 10)
vyatta@R1:~$
```

show history

Displays command execution history.

Syntax

show history [*num* | **brief**]

Command Mode

Operational mode.

Parameters

num	Displays the most recent *num* commands.
brief	Displays the most recent 20 commands.

Default

The complete command history is displayed.

Usage Guidelines

Use this command to view the history of command execution on the system. If more than one screen of output is available the ":" prompt will appear. Press the <Space> key to display the next screen, the <Enter> key to display the next line, or "q" stop the output.

Examples

Example 3-26 shows history of command execution on R1.

Example 3-26 Displaying command history

```
vyatta@R1:~$ show history
    1  2009-08-05T22:01:33+0000 configure
    2  2009-08-05T22:02:03+0000 commit
    3  2009-08-05T22:02:09+0000 exit
    4  2009-08-05T22:02:09+0000 exit
    5  2009-08-05T22:02:12+0000 exit
    6  2009-08-05T22:11:51+0000 show version
    7  2009-08-05T22:11:55+0000 configure
    8  2009-08-05T22:01:33+0000 configure
    9  2009-08-05T22:02:03+0000 commit
   10  2009-08-05T22:02:09+0000 exit
   11  2009-08-05T22:02:09+0000 exit
   12  2009-08-05T22:02:12+0000 exit
   13  2009-08-05T22:11:51+0000 show version
   14  2009-08-05T22:11:55+0000 configure
   15  2009-08-05T22:11:59+0000 show
   16  2009-08-05T22:12:27+0000 show
   17  2009-08-05T22:13:01+0000 set interfaces ethernet eth0 address
192.168.1.72/24
   18  2009-08-05T22:13:12+0000 set service ssh
   19  2009-08-05T22:13:33+0000 set system name-server 192.168.1.254
   20  2009-08-05T22:13:45+0000 set system gateway-address 192.168.1.254
   21  2009-08-05T22:13:58+0000 commit
   22  2009-08-06T05:14:15+0000 show
:
vyatta@R1:~$
```

show host

Displays host information for hosts reachable by the system.

Syntax

show host {**lookup** *hostname* | **lookup** *ipv4* | **name** | **date** | **os**}

Command Mode

Operational mode.

Parameters

lookup *hostname*	Shows the canonical name and IP address plus any configured aliases recorded in the name server for the host with the specified name.
lookup *ipv4*	Shows the canonical name and IP address plus any configured aliases recorded in the name server for the host with the specified IP address.
date	Shows the date and time according to the system clock.
name	Shows the name configured for this system.
os	Shows details about the system's operating system.

Default

None.

Usage Guidelines

Use this command to view information configured for the host.

Examples

Example 3-27 shows host information for R2.

Example 3-27 Looking up network hosts

```
vyatta@R1:~$ show host lookup R2
```

```
R2.vyatta.com        A      10.1.0.3
vyatta@R1:~$
```

Example 3-28 shows the name configured for R1.

Example 3-28 Showing network host names

```
vyatta@R1:~$ show host name
R1
vyatta@R1:~$
```

Example 3-29 shows the date and time according to the system clock.

Example 3-29 Showing the system date and time

```
vyatta@R1:~$ show host date
Mon Jan 21 17:28:47 PST 2008
vyatta@R1:~$
```

Example 3-30 shows information about the operating system.

Example 3-30 Showing operating system information

```
vyatta@R1:~$ show host os
Linux R1 2.6.23-1-486-vyatta #1 SMP Tue Jan 15 02:00:31 PST 2008 i686
GNU/Linux
vyatta@R1:~$
```

show interfaces

Displays information about system interfaces.

Syntax

show interfaces [**counters** | **detail** | **system** [**enabled**]]

Command Mode

Operational mode.

Parameters

counters	Displays summary information about all the interfaces available on your system.
detail	Displays deatailed information about all the interfaces available on your system.
system	Displays all the physical interfaces available on your system.
enabled	Shows only enabled system interfaces known to the operating system kernel.

Default

Displays information for all interfaces configured on the system.

Usage Guidelines

Use this command to view configuration information and operational status for interfaces and vifs.

When used with no option, this statement displays information for all interfaces configured on the system. You can see specific information by using other versions of this command:

To see all the physical interfaces known to the operating system kernel, use the **system** option. This option differs from the other versions of this command: the other versions show interfaces that have been configured on the system, while the **system** option shows all the physical interfaces available on your system (that is, the physical interfaces known to the operating system kernel).

The physical interfaces available to you determine which interfaces you will be able to configure and view, because you cannot configure or view an interface that does not physically exist on the system.

Examples

Example 3-32 shows the output for **show interfaces**.

Example 3-31 Displaying interface information

```
vyatta@R1:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface    IP Address                        S/L  Description
---------    ----------                        ---  -----------
eth0         -                                 u/u
eth1         192.168.74.128/24                 u/D
             2001:470:1f04:183f::2/64
eth2         172.16.139.128/24                 u/u
eth3         192.168.249.128/24                A/D
lo           127.0.0.1/8                       u/u
             ::1/128
```

Example 3-32 shows the first screen of output for **show interfaces system enabled**.

Example 3-32 Displaying detailed interface information

```
vyatta@R1:~$ show interfaces system enabled
eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 100
    link/ether 00:30:48:82:e2:0c brd ff:ff:ff:ff:ff:ff
    inet 10.1.0.54/24 brd 10.1.0.255 scope global eth0
    inet6 fe80::230:48ff:fe82:e20c/64 scope link
       valid_lft forever preferred_lft forever

    RX:  bytes    packets     errors    dropped    overrun      mcast
        348646       4144          0          0          0          0
    TX:  bytes    packets     errors    dropped    carrier collisions
        168294       1594          0          0          0          0

eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 10
    link/ether 00:30:48:82:e2:0d brd ff:ff:ff:ff:ff:ff
    inet 172.16.215.2/24 brd 172.16.215.255 scope global eth1
    inet6 fe80::230:48ff:fe82:e20d/64 scope link
       valid_lft forever preferred_lft forever

    RX:  bytes    packets     errors    dropped    overrun      mcast
```

```
          1384         11          0          0          0          0
    TX:  bytes    packets     errors    dropped    carrier collisions
          1990         18          0          0          0          0

eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
lines 1-23
```

show license

Displays Vyatta license information.

Syntax

show license

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to view Vyatta license informaiton.

Examples

Example 3-33 shows the first screen of output for **show license**.

Example 3-33 Displaying license information

```
vyatta@R1:~$ show license
GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
                  51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
```

```
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.  (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.)  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

  To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.
```

show ntp

Shows the status of configured NTP servers.

Syntax

show ntp {*host* | *ipv4* | **0.vyatta.pool.ntp.org**}

Command Mode

Operational mode.

Parameters

host	Shows the status of the connection to the NTP server with the specified host name.
ipv4	Shows the status of the connection to the NTP server at the specified IPv4 address.
0.vyatta.pool.ntp.org	Shows the status of the connection to the default NTP server.

Default

None.

Usage Guidelines

Use this command to view the status of connections to configured NTP servers.

A line entry is given for each configured NTP server, showing the server's IP address and how often the system is polling and updating to the NTP clock. An asterisk (*) next to the NTP server's IP address indicates successful synchronization with the NTP server.

NTP server connections are configured using system ntp server <name> command.

Examples

Example 3-34 shows the configured NTP server (in this case 69.59.150.135).

Example 3-34 Showing configured NTP servers

```
vyatta@R1:~$ show ntp
remote                local      st poll reach  delay   offset    disp
=======================================================================
=69.59.150.135   192.168.1.92     3   64     1 0.04057 -0.281460 0.96825
vyatta@R1:~$
```

Example 3-35 shows the NTP server at IP address 69.59.150.135.

Example 3-35 Showing information for a specific NTP server

```
vyatta@R1:~$ show ntp 69.59.150.135
server 69.59.150.135, stratum 3, offset 46.614524, delay 0.03207
22 Jan 12:20:36 ntpdate[10192]: step time server 69.59.150.135 offset
46.614524 sec
vyatta@R1:~$
```

show reboot

Shows the next scheduled reboot date and time.

Syntax

show reboot

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to view the next scheduled reboot date and time.

Examples

Example 3-36 shows the next scheduled reboot date and time.

Example 3-36 Showing the next scheduled reboot

```
vyatta@R1:~$ show reboot
Reboot scheduled for [Sat Dec 12 20:23:00 2009]
vyatta@R1:~$
```

Example 3-37 shows no scheduled reboot.

Example 3-37 Showing no scheduled reboot

```
vyatta@R1:~$ show reboot
No reboot currently scheduled
vyatta@R1:~$
```

show system boot-messages

Displays boot messages generated by the kernel.

Syntax

show system boot-messages [**all**]

Command Mode

Operational mode.

Parameters

all	Displays all kernel boot messages.

Default

A subset of the full list of kernel boot messages is displayed.

Usage Guidelines

Use this command to see startup messages that have been generated by the kernel.

Examples

Example 3-38 shows the first screen of output for **show system boot-messages**.

Example 3-38 Displaying startup messages

```
vyatta@R1:~$ show system boot-messages
Linux version 2.6.23-1-486-vyatta (autobuild@sydney) (gcc version 4.2.3
20071123 (prerelease) (Debian 4.2.2-4)) #1 SMP Fri Jan 18 07:17:50 PST 2008
BIOS-provided physical RAM map:
 BIOS-e820: 0000000000000000 - 000000000009f800 (usable)
 BIOS-e820: 000000000009f800 - 00000000000a0000 (reserved)
 BIOS-e820: 00000000000f0000 - 0000000000100000 (reserved)
 BIOS-e820: 0000000000100000 - 000000001fee0000 (usable)
 BIOS-e820: 000000001fee0000 - 000000001fee3000 (ACPI NVS)
 BIOS-e820: 000000001fee3000 - 000000001fef0000 (ACPI data)
 BIOS-e820: 000000001fef0000 - 000000001ff00000 (reserved)
 BIOS-e820: 00000000fec00000 - 0000000100000000 (reserved)
0MB HIGHMEM available.
```

```
510MB LOWMEM available.
found SMP MP-table at 000f5a20
Entering add_active_range(0, 0, 130784) 0 entries of 256 used
Zone PFN ranges:
  DMA             0 ->     4096
  Normal       4096 ->   130784
  HighMem    130784 ->   130784
Movable zone start PFN for each node
early_node_map[1] active PFN ranges
    0:        0 ->   130784
On node 0 totalpages: 130784
:
```

show system connections

Displays active network connections on the system.

Syntax

show system connections

Command Mode

Operational mode.

Parameters

None.

Default

None:

Usage Guidelines

Use this command to see what network connections are currently active on the network.

Examples

Example 3-39 shows the first screen of output for **show system connections**.

Example 3-39 Displaying active connections

```
vyatta@R1:~$ show system connections
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:179             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 192.168.1.77:22         192.168.1.102:2449      ESTABLISHED
tcp6       0      0 :::2606                 :::*                    LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::179                  :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
udp        0      0 192.168.1.77:123        0.0.0.0:*
udp        0      0 127.0.0.1:123           0.0.0.0:*
udp        0      0 0.0.0.0:123             0.0.0.0:*
```

```
udp6       0      0 fe80::20c:29ff:fe68:123 :::*
udp6       0      0 ::1:123                 :::*
udp6       0      0 :::123                  :::*
raw6       0      0 :::58                   :::*                    7
raw6       0      0 :::89                   :::*                    7
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  12     [ ]         DGRAM                    10203    /dev/log
unix  2     [ ACC ]    STREAM    LISTENING    10657   /var/run/vyatta/quagga/zserv.api
unix  2      [ ACC ]     STREAM     LISTENING     10665    /var/run/vyatta/quagg
:
```

show system kernel-messages

Displays messages in the kernel ring buffer.

Syntax

show system kernel-messages

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to see messages currently residing in the kernel ring buffer.

Examples

Example 3-40 shows the first screen of output for **show system kernel-messages**.

Example 3-40 Displaying messages from the kernel

```
vyatta@R1:~$ show system kernel-messages
Linux version 2.6.16 (autobuild@phuket.vyatta.com) (gcc version 4.1.1) #1
Tue Dec 5 15:56:41 PST 2006
BIOS-provided physical RAM map:
 BIOS-e820: 0000000000000000 - 000000000009f800 (usable)
 BIOS-e820: 000000000009f800 - 00000000000a0000 (reserved)
 BIOS-e820: 00000000000f0000 - 0000000000100000 (reserved)
 BIOS-e820: 0000000000100000 - 000000000fee0000 (usable)
 BIOS-e820: 000000000fee0000 - 000000000fee3000 (ACPI NVS)
 BIOS-e820: 000000000fee3000 - 000000000fef0000 (ACPI data)
 BIOS-e820: 000000000fef0000 - 000000000ff00000 (reserved)
 BIOS-e820: 00000000fec00000 - 0000000100000000 (reserved)
0MB HIGHMEM available.
254MB LOWMEM available.
```

```
found SMP MP-table at 000f5a20
On node 0 totalpages: 65248
  DMA zone: 4096 pages, LIFO batch:0
  DMA32 zone: 0 pages, LIFO batch:0
  Normal zone: 61152 pages, LIFO batch:15
  HighMem zone: 0 pages, LIFO batch:0
DMI 2.3 present.
Intel MultiProcessor Specification v1.4
    Virtual Wire compatibility mode.
OEM ID: OEM00000 Product ID: PROD00000000 APIC at: 0xFEE00000
:
```

show system memory

Displays system memory usage.

Syntax

show system memory [**cache** | **detail** | **quagga**]

Command Mode

Operational mode.

Parameters

cache	Displays memory cache details.
detail	Displays memory usage details.
quagga	Displays memory usage by the Quagga subsystem.

Default

None.

Usage Guidelines

Use this command to see how much memory is currently being used by the system, and how much is free.

Examples

Example 3-41 shows information about memory usage on R1.

Example 3-41 Displaying information about memory usage

```
vyatta@R1:~$ show system memory
total       used       free     shared    buffers     cached
Mem:        242836     170796      72040          0      58844      81748
Swap:            0          0          0
Total:      242836     170796      72040
vyatta@R1:~$
```

Example 3-42 shows detailed information about memory usage on R1.

Example 3-42 Displaying detailed information about memory usage

```
vyatta@R1:~$ show system memory detail
MemTotal:          242836 kB
MemFree:            72040 kB
Buffers:            58844 kB
Cached:             81760 kB
SwapCached:             0 kB
Active:             75496 kB
Inactive:           79252 kB
Active(anon):       14344 kB
Inactive(anon):       264 kB
Active(file):       61152 kB
Inactive(file):     78988 kB
Unevictable:            0 kB
Mlocked:                0 kB
HighTotal:              0 kB
HighFree:               0 kB
LowTotal:          242836 kB
LowFree:            72040 kB
SwapTotal:              0 kB
SwapFree:               0 kB
Dirty:                  0 kB
Writeback:              0 kB
AnonPages:          14172 kB
Mapped:              7464 kB
:
```

Example 3-43 shows information about memory cache usage on R1.

Example 3-43 Displaying information about memory cache usage

```
vyatta@R1:~$ show system memory cache
 Active / Total Objects (% used)    : 99681 / 100958 (98.7%)
 Active / Total Slabs (% used)      : 2690 / 2690 (100.0%)
 Active / Total Caches (% used)     : 61 / 72 (84.7%)
 Active / Total Size (% used)       : 12081.72K / 12346.32K (97.9%)
 Minimum / Average / Maximum Object : 0.01K / 0.12K / 8.00K

  OBJS ACTIVE  USE OBJ SIZE  SLABS OBJ/SLAB CACHE SIZE NAME
 30806  30806 100%    0.05K    422       73      1688K buffer_head
 19200  19178  99%    0.13K    640       30      2560K dentry
  9010   8954  99%    0.05K    106       85       424K sysfs_dir_cache
  7168   7054  98%    0.01K     14      512        56K kmalloc-8
  4864   4853  99%    0.02K     19      256        76K kmalloc-16
  2816   2693  95%    0.03K     22      128        88K kmalloc-32
  2640   2640 100%    0.38K    264       10      1056K unionfs_inode_cache
  2380   2213  92%    0.02K     14      170        56K anon_vma_chain
  2322   2322 100%    0.44K    258        9      1032K squashfs_inode_cache
  2255   2248  99%    0.34K    205       11       820K inode_cache
  2210   2199  99%    0.05K     26       85       104K ext3_xattr
  1886   1884  99%    0.09K     41       46       164K vm_area_struct
  1664   1512  90%    0.12K     52       32       208K kmalloc-128
  1536   1470  95%    0.06K     24       64        96K kmalloc-64
  1536   1433  93%    0.02K      6      256        24K anon_vma
  1313   1308  99%    0.29K    101       13       404K radix_tree_node
:
```

show system processes

Displays active system processes.

Syntax

show system processes [**extensive** | **summary** | **tree**]

Command Mode

Operational mode.

Parameters

extensive	Shows all processes and extensive details about each.
summary	Shows a summary of system usage.
tree	Shows all processes and how they are related.

Default

Lists all processes currently running on the system.

Usage Guidelines

Use this command to see information about processes currently running on the system.

Examples

Example 3-44 shows the first screen of output for **show system processes**.

Example 3-44 Displaying process information

```
vyatta@R1:~$ show system processes
PID TTY      STAT   TIME COMMAND
    1 ?        Ss     0:03 init [2]
    2 ?        S      0:00 [kthreadd]
    3 ?        S      0:00 [ksoftirqd/0]
    4 ?        S      0:00 [migration/0]
    5 ?        S      0:00 [watchdog/0]
    6 ?        S      0:09 [events/0]
    7 ?        S      0:00 [khelper]
   12 ?        S      0:00 [async/mgr]
   13 ?        S      0:00 [pm]
   99 ?        S      0:00 [sync_supers]
  101 ?        S      0:00 [bdi-default]
  102 ?        S      0:00 [kintegrityd/0]
  104 ?        S      0:00 [kblockd/0]
  106 ?        S      0:00 [kacpid]
  107 ?        S      0:00 [kacpi_notify]
  108 ?        S      0:00 [kacpi_hotplug]
  174 ?        S      0:00 [khubd]
  177 ?        S      0:00 [kseriod]
  299 ?        S      0:00 [khungtaskd]
  300 ?        S      0:00 [kswapd0]
  353 ?        S      0:00 [aio/0]
  361 ?        S      0:00 [unionfs_siod/0]
:
```

Example 3-45 shows the first screen of output for **show system processes extensive**.

Example 3-45 Displaying extensive process information

```
vyatta@R1:~$ show system processes extensive
top - 08:23:47 up 13:28,  2 users,  load average: 0.12, 0.03, 0.01
Tasks:  72 total,   1 running,  71 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0%us,  0.2%sy,  0.0%ni, 99.8%id,  0.0%wa,  0.0%hi,  0.0%si,
0.0%st
Mem:    242836k total,   170488k used,    72348k free,    58752k buffers
Swap:        0k total,        0k used,        0k free,    81440k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 3515 vyatta    20   0  2372  984  768 R  1.8  0.4   0:00.06 top
    1 root      20   0  2076  680  584 S  0.0  0.3   0:03.79 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S  0.0  0.0   0:00.98 ksoftirqd/0
    4 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/0
    5 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 watchdog/0
    6 root      20   0     0    0    0 S  0.0  0.0   0:09.69 events/0
    7 root      20   0     0    0    0 S  0.0  0.0   0:00.00 khelper
   12 root      20   0     0    0    0 S  0.0  0.0   0:00.00 async/mgr
   13 root      20   0     0    0    0 S  0.0  0.0   0:00.00 pm
   99 root      20   0     0    0    0 S  0.0  0.0   0:00.12 sync_supers
  101 root      20   0     0    0    0 S  0.0  0.0   0:00.27 bdi-default
  102 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kintegrityd/0
  104 root      20   0     0    0    0 S  0.0  0.0   0:00.05 kblockd/0
  106 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kacpid
  107 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kacpi_notify
:
```

Example 3-46 shows the first screen of output for **show system processes tree.**

Example 3-46 Displaying hierarchical process information

```
vyatta@R1:~$ show system processes tree
PID  PGID   SID TTY          TIME CMD
    2     0     0 ?        00:00:00 kthreadd
    3     0     0 ?        00:00:00   ksoftirqd/0
    4     0     0 ?        00:00:00   migration/0
    5     0     0 ?        00:00:00   watchdog/0
    6     0     0 ?        00:00:09   events/0
    7     0     0 ?        00:00:00   khelper
   12     0     0 ?        00:00:00   async/mgr
   13     0     0 ?        00:00:00   pm
   99     0     0 ?        00:00:00   sync_supers
  101     0     0 ?        00:00:00   bdi-default
  102     0     0 ?        00:00:00   kintegrityd/0
  104     0     0 ?        00:00:00   kblockd/0
  106     0     0 ?        00:00:00   kacpid
  107     0     0 ?        00:00:00   kacpi_notify
  108     0     0 ?        00:00:00   kacpi_hotplug
  174     0     0 ?        00:00:00   khubd
  177     0     0 ?        00:00:00   kseriod
  299     0     0 ?        00:00:00   khungtaskd
  300     0     0 ?        00:00:00   kswapd0
  353     0     0 ?        00:00:00   aio/0
  361     0     0 ?        00:00:00   unionfs_siod/0
  363     0     0 ?        00:00:00   crypto/0
:
```

show system routing-daemons

Displays active routing daemons.

Syntax

show system routing-daemons

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display a list of active routing daemons.

Examples

Example 3-47 shows output for **show system routing-daemons.**

Example 3-47 Displaying a list of active routing daemons

```
vyatta@R1:~$ show system routing-daemons
 zebra ripd ripngd ospfd ospf6d bgpd
```

show system storage

Displays system file system usage and available storage space.

Syntax

show system storage

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to see how much storage space is currently being used by the system, and how much is free.

Examples

Example 3-48 shows file system usage information for R1.

Example 3-48 Displaying file system and storage information

```
vyatta@R1:~$ show system storage
Filesystem            Size  Used Avail Use% Mounted on
rootfs                953M  287M  618M  32% /
udev                   10M   28K   10M   1% /dev
/dev/hda1             953M  287M  618M  32% /
/dev/hda1             953M  287M  618M  32% /dev/.static/dev
tmpfs                 126M  4.0K  126M   1% /dev/shm
/dev/hda2             9.7M  1.5M  7.8M  17% /config
vyatta@R1:~$
```

show system uptime

Displays information on how long the system has been running.

Syntax

show system uptime

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to see how long the system has been running, the number of users currently logged in, and the average system load.

Examples

Example 3-49 shows file system usage information for R1.

Example 3-49 Displaying file system and storage information

```
vyatta@R1:~$ show system uptime
20:45:59 up  3:04,  2 users,  load average: 0.00, 0.00, 0.00
vyatta@R1:~$
```

show system usb

Displays information about peripherals connected to the USB bus.

Syntax

show system usb

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to see which peripherals are connected to the USB bus.

Examples

Example 3-50 shows system USB information for R1.

Example 3-50 Displaying USB peripheral information

```
vyatta@R1:~$ show system usb
Bus 001 Device 002: ID 0d49:7212 Maxtor
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
vyatta@R1:~$
```

show tech-support

Provides a consolidated report of system information.

Syntax

show tech-support [**brief**] [**save** [*filename*] | **save-uncompressed** [*filename*]]

Command Mode

Operational mode.

Parameters

brief	Displays a summary of "**show version**", "**show configuration**", "**show interfaces**", "**show ip route**", and "**show log**" commands.
save	Saves the support information to a compressed (.gz) file. The file name takes the format **hostname.tech-support.timestamp.gz**,where *hostname* is the host name configured for the Vyatta device and *timestamp* is the time the file was saved in the format *YYYY-MM-DD-hhmmss*. For local files, a rotation mechanism is used to limit the number of output files to 100; that is, creating a 101st file causes the oldest file to be deleted.
save-uncompressed	Saves the support information to an uncompressed file. The file name takes the format **hostname.tech-support.timestamp**,where *hostname* is the host name configured for the Vyatta device and *timestamp* is the time the file was saved in the format *YYYY-MM-DD-hhmmss*. For local files, a rotation mechanism is used to limit the number of output files to 100; that is, creating a 101st file causes the oldest file to be deleted.
filename	The name of the file to save the support information to. See "Usage Guidelines" below for further details.

Default

Information is sent to the console.

Usage Guidelines

Use this command to list a technical report providing consolidated information about system components and configuration.

NOTE *Only 'admin' level users can run the command.*

This information is valuable for debugging and diagnosing system issues. You should provide the technical report whenever you open a case with Vyatta technical support.

Tech support information can be saved to a hard disk (including a Flash disk or USB device), an FTP server, or an SCP server.

The default local tech support directory is **/config/support.**

If the *filename* is specified, the support information is saved to the file **filename.hostname.tech-support.timestamp**,where *hostname* is the host name configured for the Vyatta device and *timestamp* is the time the file was saved.

If an absolute path is prepended to filename, the file is saved in that location. Otherwise, the file is saved to a location relative to the default path, which is **/config/support** directory. An FTP or SCP server can also be specified.

The following table shows the syntax for file specification for different file locations.

Table 3-2 Specifying locations for the file

Location	Specification
An absolute path	Use standard UNIX file specification.
A relative path	Specify the path name relative to the default directory.
FTP server	Use the following syntax for *filename*: ftp://*<user>:<passwd>@<host>/<file>* where *user* is the user name on the host, *passwd* is the password associated with the user name, *host* is the host name or IP address of the FTP server, and *file* is the file name, including the path.
SCP server	Use the following syntax for *filename*: scp://*<user>:<passwd>@<host>/<file>* where *user* is the user name on the host, *passwd* is the password associated with the user name, *host* is the host name or IP address of the SCP server, and *file* is the file name, including the path.

Examples

Example 3-51 shows the first screen of a technical report.

Example 3-51 Displaying consolidated system information

```
vyatta@R1:~$ show tech-support
----------------
Show Tech-Support
----------------

----------------
CONFIGURATION
----------------

----------------
Vyatta Version and Package Changes
----------------
Version:      999.larkspurse.06200031
Description:  999.larkspurse.06200031
Copyright:    2006-2010 Vyatta, Inc.
Built by:     autobuild@vyatta.com
Built on:     Sun Jun 20 07:31:17 UTC 2010
Build ID:     1006200731-27ea461
Boot via:     image
Uptime:       16:28:05 up  9:56,  1 user,  load average: 0.00, 0.00, 0.00

----------------
Configuration File
----------------
interfaces {
    ethernet eth0 {
        address 192.168.1.82/24
        duplex auto
:
```

show version

Displays information about the version of system software.

Syntax

show version [**all** | **added** | **deleted** | **downgraded** | **quagga** | **upgraded**]

Command Mode

Operational mode.

Parameters

all	Show all packages that have been added, deleted, downgraded, or upgraded since the last baseline version upgrade.
added	Show all packages that have been upgraded since the last baseline version upgrade.
deleted	Show all packages that have been deleted since the last baseline version upgrade.
downgraded	Show all packages that have been downgraded since the last baseline version upgrade.
quagga	Shows the version of quagga code used in the system.
upgraded	Show all packages that have been upgraded since the last baseline version upgrade.

Default

A brief summary of version information is shown. Detailed information about constituent packages is not shown.

Usage Guidelines

Use this command to see what package changes have occurred since the last time a full version upgrade was performed.

The information shown is always with respect to the last full version upgrade. Therefore, for example:

- Immediately following a full version upgrade, issuing a **show version all** command will show no changes.
- If a package is added after upgrading, issuing a **show version all** will show the added package.
- However, if the added package is then deleted again, issuing a **show version all** will show no change, since the system is now in the same state as immediately after the full version upgrade.

Keep in mind that if you delete a package, and packages depending on the deleted package are also removed.

Examples

Example 3-52 shows sample output for the **show version** command used with no option.

Example 3-52 Displaying a summary of version information

```
vyatta@vyatta:~$ show version
Version  :    888.islavista
Copyright:    2006-2008 Vyatta, Inc.
Built by :    root@vyatta.com
Built on :    Tue Oct 28 11:25:54 UTC 2008
Build ID :    2008-10-28-0749-f64e188
Boot via :    livecd
Uptime   :   01:29:58 up  1:30,  2 users,  load average: 0.00, 0.00, 0.00
vyatta@vyatta:~$
```

Example 3-53 shows the first page of sample output for the **show version all** command.

Example 3-53 Displaying software package version information

```
vyatta@vyatta:~$ show version all
Version  :    888.islavista
Copyright:    2006-2008 Vyatta, Inc.
Built by :    root@vyatta.com
Built on :    Tue Oct 28 11:25:54 UTC 2008
Build ID :    2008-10-28-0749-f64e188
Boot via :    livecd
Uptime   :   01:29:58 up  1:30,  2 users,  load average: 0.00, 0.00, 0.00

ADDED:
Aii aptitude 0.4.4-4
Aii libc6 2.3.6.ds1-13
Aii libdb4.4 4.4.20-8
Aii libexpat1 1.95.8-3.4
Aii libncurses5 5.5-5
Aii libnetaddr-ip-perl 3.14-2
Aii libpam0g 0.79-4
Aii libsasl2 2.1.22.dfsg1-8
Aii libtasn1-3 0.3.6-2
Aii libwrap0 7.6.dbs-13
Aii snmp 5.2.3-7
Aii supported-version 2.2
:
```

Example 3-54 shows sample output for the **show version added** command.

Example 3-54 Displaying information about added software packages

```
vyatta@vyatta:~$ show version added
Version  :    888.islavista
Copyright:    2006-2008 Vyatta, Inc.
Built by :    root@vyatta.com
Built on :    Tue Oct 28 11:25:54 UTC 2008
Build ID :    2008-10-28-0749-f64e188
Boot via :    livecd
Uptime   :   01:29:58 up  1:30,  2 users,  load average: 0.00, 0.00, 0.00

ADDED:
Aii aptitude 0.4.4-4
Aii libc6 2.3.6.ds1-13
Aii libdb4.4 4.4.20-8
Aii libexpat1 1.95.8-3.4
Aii libncurses5 5.5-5
Aii libnetaddr-ip-perl 3.14-2
Aii libpam0g 0.79-4
Aii libsasl2 2.1.22.dfsg1-8
Aii libtasn1-3 0.3.6-2
Aii libwrap0 7.6.dbs-13
Aii snmp 5.2.3-7
Aii supported-version 2.2
Aii sysvinit 2.86.ds1-38
Aii tasksel 2.66
Aii vyatta-bgp 1.4-9
Aii vyatta-cli 2.1.1-9
Aii vyatta-config-migrate 2.1.1-4
Aii vyatta-dhcp-support 2.1.1-4
Aii vyatta-firewall 1.4-9
Aii vyatta-nat 2.1.1-5
Aii vyatta-nat-cli 2.1.1-4
Aii vyatta-nat-xorp 2.1.1-3
Aii vyatta-ospf 1.4-9
Aii vyatta-rip 1.4-9
Aii vyatta-xg 1.4-9
Aii zlib1g 1.2.3-13
:
```

system console device <device>

Defines the specified device as a system console.

Syntax

set system console device *device* [**speed** *speed*] [**modem**]

delete system console device *device* [**speed**] [**modem**]

show system console device *device*

Command Mode

Configuration mode.

Configuration Statement

```
system {
    console {
        device device {
            speed speed
            modem
        }
    }
}
```

Parameters

device	Multi-node. The name of the console device (e.g. **ttyS0**).
speed	The speed (baudrate) of the console device. Supported values are: **1200**, **2400**, **4800**, **9600**. **19200**, **38400**, **57600**, and **115200**. The default is **9600**.
modem	Indicates that the port is connected to the serial console via a Hayes compatible modem.

Default

Device **ttyS0** is configured with a speed of **9600**.

Usage Guidelines

Use this command to specify configuration for a sytem console.

Changes take effect the next time a user logs on via the device, NOT when the configuration is **commit**ted.

Standard VGA consoles (tty0-tty9) always exist and are not controlled by this configuration.

Bootup messages are limited to ttyS0. Other consoles can be configured but will not receive these messages.

Changing the speed of serial devices does not affect the system BIOS.

Use the **set** form of this command to specify configuration for a sytem console.

Use the **delete** form of this command to remove a system console device.

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

system console powersave

Enables blank screen powersaving on VGA console screens.

Syntax

set system console powersave

delete system console powersave

show system console

Command Mode

Configuration mode.

Configuration Statement

```
system {
    console {
        powersave
    }
}
```

Parameters

None.

Default

Blank screen powersaving is not enabled on a VGA console.

Usage Guidelines

Use this command to enable blank screen powersaving on a VGA console. After 15 minutes of inactivity the screen will go blank. After 60 minutes the monitor will power down.

Use the **set** form of this command to enable blank screen powersaving on a VGA console.

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

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

system domain-name <domain>

Sets the system's domain.

Syntax

set system domain-name *domain*

delete system domain-name

show system domain-name

Command Mode

Configuration mode.

Configuration Statement

```
system {
   domain-name domain
}
```

Parameters

domain	Mandatory. The domain where the system resides; for example, "vyatta.com". The format is a string containing letters, numbers, hyphens ("-") and one period. The domain name can be a maximum of 253 characters.

Default

None.

Usage Guidelines

Use this command to set the system's domain.

Note that both **domain-name** and **domain-search** cannot be configured simultaneously - they are mutually exclusive.

Use the **set** form of this command to specify the domain name to be used by the system.

Use the **delete** form of this command to remove the domain name.

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

system domain-search domain <domain>

Defines a set of domains for domain completion.

Syntax

set system domain-search domain *domain*

delete system domain-search domain *domain*

show system domain-search domain

Command Mode

Configuration mode.

Configuration Statement

```
system {
   domain-search {
      domain domain
   }
}
```

Parameters

domain	Mandatory. Multi-node. The domain name to be added to or deleted from the list of domains in the search order string. The format is a string specifying a domain; for example **vyatta.com**. Letters, numbers, hyphens ("-") and one period (".") are allowed. The domain name can be a maximum of 253 characters. You can specify up to 6 domains by creating up to 6 **domain-search** multi-nodes.

Default

None.

Usage Guidelines

Use this command to list up to 6 domains to be searched in DNS lookup requests.

When the system receives an unqualified host name, it attempts to form a Fully Qualified Domain Name (FQDN) by appending the domains in this list to the host name. The system tries each domain name in turn, in the order in which they were configured. If none of the resulting FQDNs succeeds, the name is not resolved and an error is reported.

Note that both **domain-name** and **domain-search** cannot be configured simultaneously - they are mutually exclusive.

Use the **set** form of this command to add a domain to the search list. Note that you cannot use **set** to change a domain name in the list. To replace an incorrect domain, delete it and replace it with a new one.

Use the **delete** form of this command to remove a domain name from the list.

Use the **show** form of this command to view the list of domain names.

system gateway-address <address>

Specifies the default gateway for the system.

Syntax

set system gateway-address *address*

delete system gateway-address

show system gateway-address

Command Mode

Configuration mode.

Configuration Statement

```
system {
   gateway-address address
}
```

Parameters

address	Mandatory. The IPv4 address of the default gateway.

Default

None.

Usage Guidelines

Use this command to set the location of the default gateway.

The default gateway is the location where packets are routed when the destination does not match any specific routing entries. Only one default gateway can be set per system.

Use the **set** form of this command to specify the address of default gateway.

Use the **delete** form of this command to remove the default gateway. Note that, in most cases, traffic cannot be routed correctly if a default gateway is not specified.

Use the **show** form of this command to view the address of the default gateway.

system host-name <name>

Sets the host name for the system.

Syntax

set system host-name *name*

delete system host-name

show system host-name

Command Mode

Configuration mode.

Configuration Statement

```
system {
   host-name name
}
```

Parameters

name	The name you want to give the system. Letters, numbers, and hyphens (“-”) only are allowed. The default is “vyatta”. If you delete the host name, or if you try to delete the **system** node, the host name reverts to the default.

Default

By default, the host name is preconfigured to “vyatta”. If you delete the host name, or if you delete the **system** node, the default values are restored.

Usage Guidelines

Use this command to specify a host name for the system.

When you set this value, the command prompt changes to reflect the new host name. To see the change in the prompt, you must log out of the system shell and log back in again.

Use the **set** form of this command to modify the host name.

Use the **delete** form of this command to restore the default host name (“vyatta”).

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

system name-server <address>

Specifies the DNS name servers available to the system.

Syntax

set system name-server *address*

delete system name-server *address*

show system name-server

Command Mode

Configuration mode.

Configuration Statement

```
system {
   name-server address {}
}
```

Parameters

address	Multi-node. The IPv4 or IPv6 address of a DNS name server to use for local name query requests. You can specify multiple DNS name servers by creating multiple instances of the **name-server** configuration node.

Default

None.

Usage Guidelines

Use this command to specify domain name servers (DNS) for the system.

Use the **set** form of this command to define a name server for the system. Note that you cannot modify a DNS name server entry using the **set** command. To replace a name server entry, delete the entry and create a new one.

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

Use the **show** form of this command to view the name servers that have been defined.

system ntp server <name>

Specifies the NTP servers to use when synchronizing the system’s clock.

Syntax

set system ntp server *server*

delete system ntp server *server*

show system ntp server

Command Mode

Configuration mode.

Configuration Statement

```
system {
    ntp {
        server server
    }
}
```

Parameters

server	Multi-node. The IP address or host name of an NTP server. The system will automatically obtain the system date and time from the specified server(s). You can specify multiple NTP servers by creating multiple instances of the **ntp server** configuration node.

Default

By default, the system uses the NTP server at **0.vyatta.pool.ntp.org**.

Usage Guidelines

Use this command to specify NTP servers for the system.

Use the **set** form of this command to specify an NTP server for the system. Note that you cannot modify an NTP server entry using the **set** command. To replace an NTP server entry, delete the entry and create a new one.

Use the **delete** form of this command to remove an NTP server.

Use the **show** form of this command to view the NTP servers that have been defined.

system options reboot-on-panic <value>

Allows you set system behavior on system panic.

Syntax

set system options reboot-on-panic *value*

delete system options reboot-on-panic

show system options reboot-on-panic

Command Mode

Configuration mode.

Configuration Statement

```
system {
   options {
      reboot-on-panic value
   }
}
```

Parameters

value	Mandatory. Indicates whether or not the system should automatically reboot if a kernel panic occurs. Supported values are as follows: **true:** The system reboots if a kernel panic occurs. **false:** The system does not reboot if a kernel panic occurs.

Default

The default is **true**.

Usage Guidelines

Configuring the system not to reboot on kernel panic allows you to examine information that might help you determine the cause of the panic.

Use the **set** form of this command to specify whether or not to reboot on kernel panic.

Use the **delete** form of this command to restore this option to its default value.

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

system static-host-mapping host-name <name>

Defines a static mapping between a host name and an IP address.

Syntax

set system static-host-mapping host-name *name* [**inet** *address* | **alias** *alias*]

delete system static-host-mapping host-name *name* [**inet** | **alias**]

show system static-host-mapping host-name *name* [**inet** | **alias**]

Command Mode

Configuration mode.

Configuration Statement

```
system {
   static-host-mapping {
      host-name name {
         inet address
         alias alias {}
      }
   }
}
```

Parameters

name	Multi-node. The Fully Qualified Domain Name (FQDN) name being statically mapped to an IP address (for example, **router1@mydomain.com**). Letters, numbers, periods (".") and hyphens ("-") only are allowed. You can define multiple mappings by creating multiple **host-name** configuration nodes.
address	Mandatory. The IPv4 address of the interface being statically mapped to the host name.
alias	Optional. Multi-node. An alias for the interface. Letters, numbers, and hyphens are allowed. You can define multiple aliases for a host by creating multiple **alias** configuration nodes.

Default

None.

Usage Guidelines

Use this command to statically map a host name to an IP address and one or more aliases.

Use the **set** form of this command to create a new static mapping between a host name and an IP address, assign an address, or specify an alias. Note that you cannot use **set** to change the host name. To change the host name, delete the mapping entry and create a new one with the correct host name.

Use the **delete** form of this command to remove a static mapping, an address, or an alias.

Use the **show** form of this command to view a static mapping, an address, or an alias.

system time-zone <zone>

Sets the time zone for the local system clock.

Syntax

set system time-zone *zone*

delete system time-zone

show system time-zone

Command Mode

Configuration mode.

Configuration Statement

```
system {
    time-zone zone
}
```

Parameters

zone	A string representing the time zone. The format is **Region/Location**. For example, **US/Pacific**. Use command completion (i.e. the <Tab> key) to display the various options.

Default

The default is **GMT** (Greenwich Mean Time).

Usage Guidelines

Use this command to set the time zone for the local system clock. To do this, you specify a Region and Location in the format **Region/Location**. Note that both Region and Location are case sensitive. Use command completion (i.e. the <Tab> key) to display the various options.

In addition to the wide range of Region/Locations available, backwards compatibility is achieved by using **Etc/<offset>** and **SystemV/<offset>** as **Region/Location**. Note that **Etc/<offset>** uses Posix-style offsets. These use positive signs to indicate west of Greenwich rather than east of Greenwich as many systems do. For example, **Etc/GMT+8** corresponds to 8 hours behind UTC (that is, west of Greenwich).

Use the **set** form of this command to set the time zone for the first time, or to change the time zone setting.

Use the **delete** form of this command to remove the time zone setting. This restores the time zone to the default (GMT).

Use the **show** form of this command to view the time zone setting.

terminal

Controls behaviors of the system terminal.

Syntax

terminal {**key query-help** {**enable**|**disable**} | **length** *length* | **pager** [*pager*] | **width** *width*}

Command Mode

Operational mode.

Parameters

key query-help	Set whether or not you can get help using a question mark. The options are **enable** and **disable**.The default is **enable**.
length	Sets the terminal screen length to a given number of rows.
pager	The program to use as the terminal pager. If none is specified, the default (less) is used.
width	Sets the terminal screen width to a given number of columns.

Default

None.

Usage Guidelines

Use this command to set the terminal behavior.

Chapter 4: User Management

This chapter explains how to set up user accounts and user authentication.

This chapter presents the following topics:

User Management Configuration

This section presents the following topics:

- User Management Overview
- Creating "Login" User Accounts
- Configuring for a RADIUS Server
- Configuring for a TACACS+ Server
- Configuring for SSH Access using Shared Public Keys

User Management Overview

This system presents the following topics:

- Login Authentication
- RADIUS Authentication
- TACACS+ Authentication
- Order of Authentication
- SSH Access using Shared Public Keys

The Vyatta System supports all of the following methods of authentication:

- Role-based user account management through a local user database ("login" authentication)
- Remote Authentication Dial In User Service (RADIUS) authentication server
- Terminal Access Control Access-Control System Plus (TACACS+) authentication server
- SSH access using a shared public key for authentication

Login Authentication

The system creates a single login user account by default: user **vyatta** with password **vyatta**. It is highly recommended that, for security reasons, this password be changed.

If no RADIUS or TACACS+ server has been configured, the system authenticates users with the password configured using the system login user <user> authentication command.

NOTE *The Vyatta system supports only local authentication for IPv6. RADIUS and TACACS+ are not supported for IPv6.*

You can change user account information using lower-level operating system commands, but changes made in this way do not persist across reboots. For persistent changes to user account information, use the Vyatta CLI.

Note that, in the Vyatta system, the Linux **passwd** command can only be used by administrative users.

The **login** configuration node is a mandatory node. It is created automatically with default information when the system is first started. If this node is subsequently deleted, the system recreates it with default information when restarted.

Login user passwords are supplied in plain text. After configuration is committed, the system encrypts them and stores the encrypted version internally. When you display user configuration, only the encrypted version of the password is displayed.

Note that the login authentication prompt has a total timeout interval of 60 seconds. The sum all timeout intervals must fall within that limit; otherwise—that is, if cumulative RADIUS and TACACS+ server timeout intervals exceed 60 seconds—the login process will time out and must be repeated.

RADIUS Authentication

RADIUS servers are used only to authenticate user passwords. Using RADIUS authentication does not affect a user's configured privilege level. RADIUS authentication is not supported for IPv6.

To configure RADIUS, you specify the location of the RADIUS server and specify the secret to be used to authenticate the user on the RADIUS server. RADIUS secrets are specified in plain text. They are stored in plain text on the system, and used as part of a cryptographic operation for transferring authentication information securely over the network. When you view RADIUS secrets, they are displayed in plain text. RADIUS secrets must not contain spaces and are case-sensitive.

Where RADIUS authentication is used, some delay can be expected; the amount of delay depends on the cumulative timeout values configured for all RADIUS servers.

If you are using RADIUS authentication, the users must still be configured in the Vyatta login database; otherwise, the user is not able to access the Vyatta system and therefore is not able to query the RADIUS server.

TACACS+ Authentication

This feature is available only in the Vyatta Subscription Edition.

This section presents the following topics:

- Using TACACS+ Alone
- Using TACACS+ and the Vyatta System Together
- Mapping Vyatta User IDs To TACACS+ Usernames
- Specifying authentication level in TACACS+
- Connection Type Information Sent to TACACS+
- Troubleshooting TACACS+ Authentication Issues

TACACS+ is a distributed access control system for routers providing authentication, authorization, and accounting. TACACS+ authentication is not supported for IPv6.

To configure TACACS+, you specify the location of the TACACS+ server and specify the secret to be used to authenticate the user on the TACACS+ server. TACACS+ secrets are specified in plain text. TACACS+ secrets are stored in plain text on the system, and used as part of a cryptographic operation for transferring authentication information securely over the network. When you view TACACS+ secrets, they are displayed in plain text. TACACS+ secrets must not contain spaces and are case-sensitive.

Where TACACS+ authentication is used, some delay can be expected as the TACACS+ server is queried; the amount of delay depends on the cumulative timeout values configured for all TACACS+ servers.

Unlike RADIUS, TACACS+ authentication does not require prior authentication in the Vyatta system's login database: a TACACS+ server can be used either as the only authentication server or as a supplement the Vyatta system, providing password authentication.

Using TACACS+ Alone

If no local login user accounts are configured, user accounts on the TACACS+ system share local system account information from the default account (**tacplus-operator**). These users are given operator-level permissions. Administrator accounts share permissions with TACACS+ administrators (**tacplus-admin**).

Using TACACS+ and the Vyatta System Together

If local login user accounts on the Vyatta system also exist with the same user name on the TACACS+ server, both systems use the TACACS+ server to provide authentication, authorization, and accounting services. In this case, system account information is obtained from the local user database but the TACACS+ server is used to authorize access. If the TACACS+ server is unavailable, the local Vyatta system is used to authorize access.

Mapping Vyatta User IDs To TACACS+ Usernames

You can map a Vyatta local user ID to a different username recorded on a TACACS+ server. The mapping is specified on the TACACS+ server.

For example, to map username **tac-user** on the TACACS+ server to username **vyatta-user** on the local Vyatta system, the (partial) configuration on the TACACS+ server would look as follows:

```
user = tac-user {
     default service = permit
     login = des "aXcnmMELgIKQQ" #vyatta
     service = vyatta-exec {
        local-user-name = "vyatta-user"
     }
}
```

Logging in to the local Vyatta system using the account ID **tac-user** would actually log the user on to the Vyatta system as **vyatta-user**.

Specifying authentication level in TACACS+

By default, TACACS+ authorized users on the Vyatta system are given operator-level access. However, you can specify the authentication level for individual TACACS+ authorized users on the local Vyatta system. Like the mapping of user IDs, this configuration is specified on the TACACS+ server, as in the following example:

```
user = administrator {
     default service = permit
     login = cleartext "vyatta"
     service = vyatta-exec {
        level = "admin"
     }
}
```

Logging in to the local Vyatta system as user **administrator** in this instance will provide admin-level access.

Connection Type Information Sent to TACACS+

The Vyatta system will send different connection type information via the TACACS+ protocol based on the type of connection the user is accessing the Vyatta system by.

Table 4-1 Protocol values sent to TACACS+ based on connection type.

Connection type	Protocol value sent to TACACS+
via console	login
via ssh	sshd
via telnet	telnet

This information can be used to restrict how certain user types are allowed to access the system. For example, it is possible to restrict administrators to only login via the physical console rather than remotely via ssh or telnet.

TROUBLESHOOTING TACACS+ AUTHENTICATION ISSUES

Because TACACS+ requires a secret, data is encrypted and therefore tdebugging authentication problems can be difficult. Tools such as **tshark** can be used, provided that the secret is known.

For example, to debug a TACACS+ authentication problem using **tshark**, given a secret of “mysecret”, on the well-known TACACS+ port (“tacacs”, which is port 43), you would either of the following commands:

```
tshark -o tacplus.key:mysecret tcp port tacacs
tshark -o tacplus.key:mysecret tcp port 43
```

Order of Authentication

By default, the system looks first for configured TACACS+ servers, then for configured RADIUS servers, and finally in the local user database. If a server configuration is found, the system queries the first configured server of that type using the configured secret. After the query is validated, the server authenticates the user from information in its database.

TACACS+ and RADIUS servers are queried in the order in which they were configured. If a query times out, the next server in the list is queried. If all queries fail, the system attempts to authenticate the user through the local Vyatta authentication database. If local authentication fails, the access attempt is rejected.

NOTE *The login process itself has a 60-second timeout value. If a user cannot be authenticated in this time by a configured authentication server then the login attempt will time out.*

If the system is configured for TACACS+ and a user is configured on it as well as on the local user database, if the user fails authentication on TACACS+ the login attempt fails. The local user database is only used in cases where the user does not exist on the TACACS+ server or the TACACS+ server becomes unavailable.

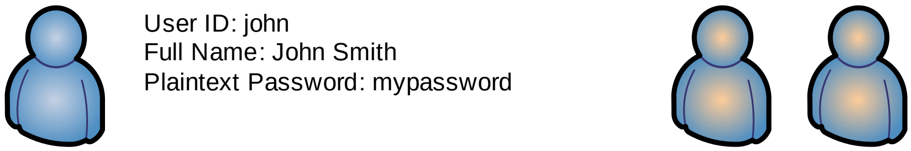
SSH Access using Shared Public Keys

Remote access to the Vyatta system is typically accomplished through either Telnet or SSH. For either of these methods, passwords can be authenticated using the local login user database, a RADIUS server, or a TACACS+ server, as described above. SSH is typically used where a secure session is required. One potential problem with password authentication, even using SSH, is that password authentication is susceptible to brute force password guessing. An alternative to password authentication, which mitigates this risk, is to authenticate SSH users using shared public keys. With this method, a private and public key pair are generated (typically using the Linux **ssh-keygen** command) on a remote system. The public key file (typically with a **.pub** extension) is loaded into the login configuration for the user that will be accessing the system with it using loadkey command. In addition, the Vyatta system must be configured to disable password authentication for SSH (see the *Vyatta Services Reference Guide*). So, SSH users can be authenticated using passwords or shared public keys, but not both.

Creating "Login" User Accounts

In this section, a sample configuration is presented for a user account that will be validated using the local user database. The sample configuration used is shown in Figure 4-1.

Figure 4-1 "Login" User Account



This section includes the following example:

- Example 4-1 Creating a "login" user account

Example 4-1 creates a user account for **John Smith**. John has a user ID of **john** and will use a plain text password of **mypassword.** Note that once configuration has been committed, only the encrypted version of the password displays when configuration is shown.

NOTE *User information can be changed through the UNIX shell (providing you have sufficient permissions). However, any changes to Vyatta router user accounts or authentication through the UNIX shell will be overwritten the next time you commit Vyatta router CLI configuration.*

To create a login user account, perform the following steps in configuration mode:

Example 4-1 Creating a “login” user account

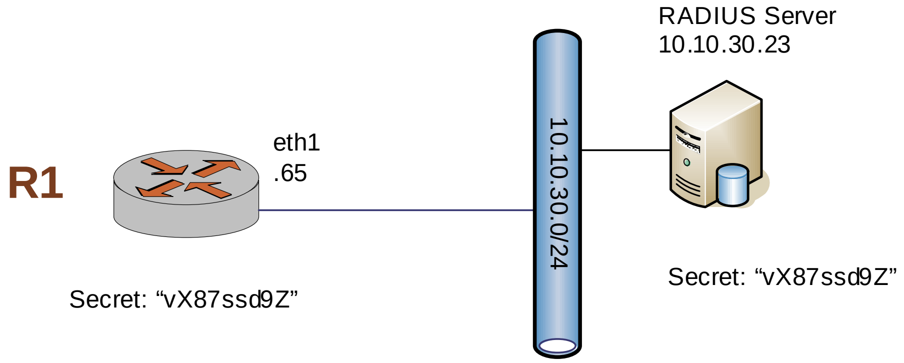
Step	Command
Create the user configuration node, define the user ID, and give the user’s full name.	`vyatta@R1#` **`set system login user john full-name “John Smith”`**
Specify the user’s password in plain text.	`vyatta@R1#` **`set system login user john authentication plaintext-password mypassword`**
Commit the change. After a password has been committed, it can be displayed only in encrypted form, as the value of the **encrypted-password** attribute.	`vyatta@R1#` **`commit`**

Show the contents of the **system login** configuration node.	`vyatta@R1#` **`show system login`**
	`user vyatta {`
	`    authentication {`
	`        encrypted-password $1$$ZbzUPUD24iyfRwCKIT16q0`
	`    }`
	`}`
	`user john {`
	`    authentication`
	`        encrypted-password $1$$Ht7gBYnxI1xCdO/JOnodh.`
	`        plaintext-password ""`
	`    }`
	`    full-name "John Smith"`
	`}`

Configuring for a RADIUS Server

This section provides a sample configuration for configuring a RADIUS authentication server, as shown below.

Figure 4-2 RADIUS Server Configuration



The example defines a RADIUS authentication server at IP address 10.10.30.23. The system is to access the RADIUS server using a secret of **vX87ssd9Z.** Configuring the server address and the secret are the minimal configuration requirements. The port and timeout values can be changed if required.

NOTE *You should take some thought selecting the shared secret, since it is this string that prevents snooping attacks on passwords. Since this value is used on every packet, it is important to choose a value that makes brute force attacks more difficult; this key should be harder to guess than any password on the system.*

To define this RADIUS server, perform the following steps in configuration mode:

Example 4-2 Configuring for a RADIUS server

Step	Command
Provide the location of the server, and the secret to be used to access it.	`vyatta@R1#` **`set system login radius-server 10.10.30.23 secret vX87ssd9Z`**
Commit the change.	`vyatta@R1#` **`commit`**

Save the configuration so that the changes persist after reboot.	`vyatta@R1#` **`save`**
	`Saving configuration to '/config/config.boot'...`
	`Done`

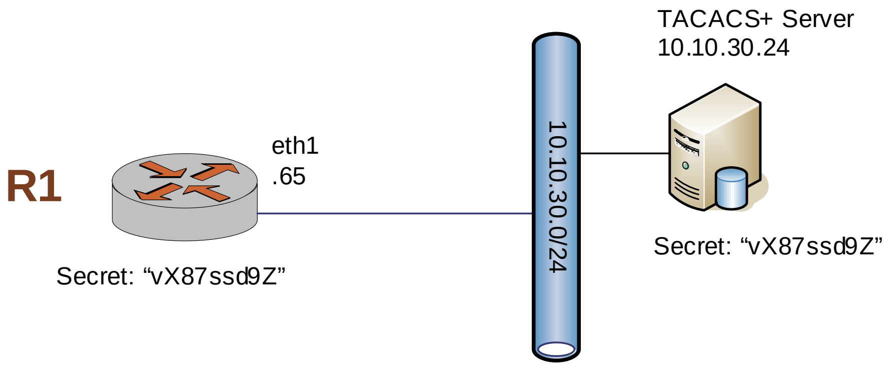
Show the contents of the **system radius-server** configuration node.	`vyatta@R1#` **`show system radius-server`**
	`radius-server 10.10.30.23 {`
	`    secret vX87ssd9Z`
	`}`

Configuring for a TACACS+ Server

This feature is available only in the Vyatta Subscription Edition.

This section provides an example of configuring for a TACACS+ authentication server, as shown below.

Figure 4-3 TACACS+ Server Configuration



The example defines a TACACS+ authentication server at IP address 10.10.30.24. The system is to access the TACACS+ server using a secret of **vX87ssd9Z.** Configuring the server address and the secret are the minimal configuration requirements. The port and timeout values can be changed if required.

NOTE *You should take some thought selecting the shared secret, since it is this string that prevents snooping attacks on passwords. Since this value is used on every packet, it is important to choose a value that makes brute force attacks more difficult; this key should be harder to guess than any password on the system*

To define a TACACS+ server, perform the following steps in configuration mode:

Example 4-3 Configuring for a TACACS+ server

Step	Command
Provide the location of the server, and the secret to be used to access it.	`vyatta@R1# set system login tacplus-server 10.10.30.24 secret vX87ssd9Z`
Commit the change.	`vyatta@R1# commit`

Save the configuration so that the changes persist after reboot.	`vyatta@R1# save`
	`Saving configuration to '/config/config.boot'...`
	`Done`

Example 4-3 Configuring for a TACACS+ server

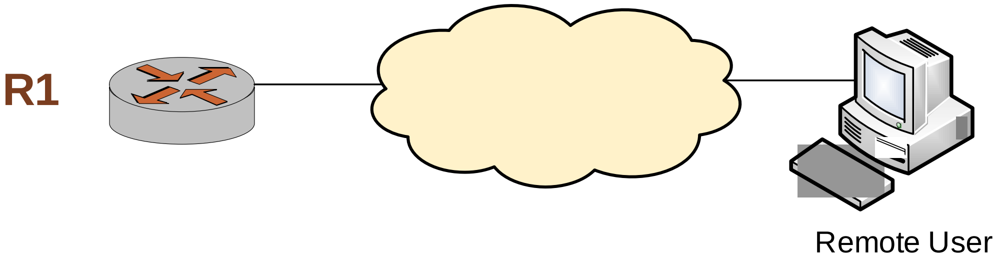
Step	Command

Show the contents of the **system tacplus-server** configuration node.	`vyatta@R1#` **`show system tacplus-server`**
	`tacplus-server 10.10.30.24 {`
	`    secret vX87ssd9Z`
	`}`

Configuring for SSH Access using Shared Public Keys

This section provides an example of configuring SSH access using shared public keys, as shown below.

Figure 4-4 SSH access using shared public keys



The example configures a Vyatta system for SSH access using shared public keys for authentication and disables password authentication (though disabling password authentication is not a prerequisite to using shared public keys for authentication). In this case the user **John Smith** (username = **john**) already exists on the system. Also, the public key (**xxx.pub**) has been previously generated (using the Linux command **ssh-keygen**) and is located in a directory owned by user **j2** on **xyz.abc.com**.

To configure for SSH access using shared public keys, perform the following steps in configuration mode:

Example 4-4 Configuring for SSH access using shared public keys

Step	Command
Set the system to disable password authentication for SSH. Note that this step is not strictly necessary but required if users are only to use shared public key authentication.	`vyatta@R1#` **`set service ssh disable-password-authentication`**

Example 4-4 Configuring for SSH access using shared public keys

Step	Command
Commit the change.	`vyatta@R1#` **`commit`**

Display the change.	`vyatta@R1#` **`show service ssh`**
	` disable-password-authentication`

Load the shared public key (**xxx.pub**) from the system where it is located and associate it with user **john**. In this case it is located on **xyz.abc.com** in a directory owned by user **j2**.	`vyatta@R1#` **`loadkey john`**
	`scp://j2@xyz.abc.com/home/j2/.ssh/xxx.pub`
	`Enter host password for user ‘j2’:`
	`################################### 100.0% Done`

Commit the change.	`vyatta@R1#` **`commit`**

Save the configuration so that the changes persist after reboot.	`vyatta@R1#` **`save`**
	`Saving configuration to '/config/config.boot'...`
	`Done`

Display the change.	`vyatta@R1#` **`show system login`**
	`user vyatta {`
	`    authentication {`
	`       encrypted-password $1$$ZbzUPUD24iyfRwCKIT16q0`
	`    }`
	`}`
	`user john {`
	`    authentication`
	`       encrypted-password $1$$Ht7gBYnxI1xCdO/JOnodh.`
	`       plaintext-password ""`
	`       public-keys j2@xyz.abc.com {`
	`             key`
	`AAAAB3NzaC1yc2EAAAABIwAAAIEAqaCtQr8hr6iUEvvQD3hGyryR5k+/UjF`
	`RFrHbqHNhjxdlYviXveVXoZrKAKHtANRp5E+j4WZMbSd4oYt9P9lFevyZv3`
	`xmdZE+ukuPlQBBAUnL29k1FtJ+G7I5tXGun9VR07JzUpEb8/KP1U4ajYClc`
	`3HxpOLpu5AU5u7jvKu/wA0=`
	`             type ssh-rsa`
	`         }`
	`    }`
	`    full-name "John Smith"`
	`}`

User Management Commands

Configuration Commands	
loadkey	Loads a shared public key for an SSH user.
show login	Displays current user's login credentials.
system login	Creates the configuration node for user management and authentication.
system login banner post-login <banner>	Specifies the post-login banner.
system login banner pre-login <banner>	Specifies the pre-login banner.
system login radius-server <address>	Defines a RADIUS server for user authentication.
system login tacplus-server <address>	Defines a TACACS+ server for user authentication.
system login user <user>	Creates a user account.
system login user <user> authentication	Sets an authentication password for a user.
system login user <user> authentication public-keys	Specifies parameters for SSH shared public key user authentication.
system login user <user> full-name <name>	Allows you to record a user's full name.
system login user <user> group <group>	Allows you to make a user a member of a group.
system login user <user> home-directory <dir>	Allows you to specify a user's home directory.
system login user <user> level <level>	Specifies a user's privilege level and system access.
system tacplus-options command-accounting	Enables logging accounting records for interactive shell commands.
Operational Commands	
show system login users	Displays user account information.
show users	Shows which users are currently logged on to the system.

loadkey

Loads a shared public key for an SSH user.

Syntax

loadkey *user file-name*

Command Mode

Configuration mode.

Configuration Statement

None.

Parameters

user	The name of the user to associate the shared public key with. The user must already be defined on the Vyatta system.
file-name	The name of the shared public key file, including the full path to its location. Shared public key files are typically generated on the remote system using the Linux **ssh-keygen** command and have a **.pub** extention. Their contents include the authentication type (for example, **ssh-rsa** or **ssh-dsa**), the key value string, and the remote system user id (for example, john@abc.com).

Default

None.

Usage Guidelines

Use this command to load a shared public key for SSH from a file into the **public-keys** configuration for a user (see "system login user <user> authentication public-keys" on page 219). This saves having to manually enter the shared public key.

NOTE *This command can only be run if there are no uncommitted changes.*

The shared public key, generated on the remote system, can be loaded from a hard disk (including a Flash disk or USB device), a TFTP server, an FTP server, an SCP server, or an HTTP server.

If a public key is loaded that contains a remote system user id that is the same as an existing **public-keys** name for a user, the existing key will be overwritten.

The following table shows the syntax for file specification for different file locations.

Table 4-2 Specifying locations for the shared public key file

Location	Specification
An absolute path on the local system	Use standard UNIX file specification.
FTP server	Use the following syntax for *file-name*: ftp://*user:passwd@host*/*key-file* where *user* is the user name on the host, *passwd* is the password associated with the user name, *host* is the host name or IP address of the FTP server, and *ckey-file* is the key file, including the path. If you do not specify *user* and *passwd* you will be prompted for them.
SCP server	Use the following syntax for *file-name*: scp://*user:passwd@host*/*key-file* where *user* is the user name on the host, *passwd* is the password associated with the user name, *host* is the host name or IP address of the SCP server, and *key-file* is the key file, including the path. If you do not specify *user* and *passwd* you will be prompted for them.
HTTP server	Use the following syntax for file-name: http://*host*/*key-fil*e where *host* is the host name or IP address of the HTTP server, and *key-file* is the key file, including the path.
TFTP server	Use the following syntax for *file-name*: tftp://*host*/*key-file* where *host* is the host name or IP address of the TFTP server, and *key-file* is the key file, including the path relative to the TFTP root directory.

show login

Displays current user’s login credentials.

Syntax

show login [groups | level | user]

Command Mode

Operational mode.

Parameters

groups	Displays the groups the user is a member of.
level	Displays the user’s login level.
user	Displays the user’s login id.

Default

Displays all credentials of the current user.

Usage Guidelines

Use this command to display credential information about the current user.

Examples

Example 4-5 shows credential information about the current user.

Example 4-5 Displaying credential information about the current user

```
vyatta@R1:~$ show login
login     : vyatta   pts/0        Aug 11 17:19 (192.168.1.150)
level     : admin
user      : vyatta
groups    : users adm disk sudo dip quaggavty vyattacfg
vyatta@R1:~$
```

show system login users

Displays user account information.

Syntax

show system login users [**all** | **locked** | **other** | **vyatta**]

Command Mode

Operational mode.

Parameters

all	Displays information about all accounts.
locked	Displays information about locked accounts.
other	Displays information about non-Vyatta accounts.
vyatta	Displays information about Vyatta accounts.

Default

Displays information about Vyatta accounts.

Usage Guidelines

Use this command to see various details about system accounts. It shows information about the last time each user logged in.

Examples

Example 4-6 shows information about Vyatta user accounts on R1.

Example 4-6 Displaying information about user accounts

```
vyatta@R1:~$ show system login users
Username        Type    Tty      From                Last login
dave            vyatta                               never logged in
test            vyatta  pts/0    192.168.1.10        Wed Mar  3 04:49:02 2010
vyatta          vyatta  pts/1    192.168.1.154       Wed Mar  3 04:59:16 2010
```

```
vyatta@R1:~$
```

show users

Shows which users are currently logged on to the system.

Syntax

show users

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to see which users are currently logged on to the system.

Examples

Example 4-7 shows information about users currently logged on to R1.

Example 4-7 Displaying information about currently logged in users

```
vyatta@R1:~$ show users
NAME     LINE         TIME         COMMENT
vyatta   tty1         Feb 22 20:58
test     pts/0        Mar  3 04:49 (192.168.1.10)
vyatta   pts/1        Mar  3 04:59 (192.168.1.154)
vyatta@R1:~$
```

system login

Creates the configuration node for user management and authentication.

Syntax

set system login

delete system login

show system login

Command Mode

Configuration mode.

Configuration Statement

```
system {
   login {}
}
```

Parameters

None.

Default

None.

Usage Guidelines

Use this command, and its sub-commands, to manage user accounts and authentication.

The **login** configuration node is a mandatory node. It is created automatically with default information when the system is first started. If this node is subsequently deleted, the system recreates it with default information.

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

Use the **delete** form of this command to restore default user information and authentication information.

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

system login banner post-login <banner>

Specifies the post-login banner.

Syntax

set system login banner post-login *banner*

delete system login banner post-login

show system login banner post-login

Command Mode

Configuration mode.

Configuration Statement

```
system {
   login {
      banner {
         post-login banner
      }
   }
}
```

Parameters

banner	The banner to be displayed during login after the user enters a valid password. The string must be enclosed in double-quotes. Special characters such as newline (\n) and tab (\t) can also be entered.

Default

The system displays operating system and copyright information.

Usage Guidelines

Use this command to specify the text that will appear when a user logs into the system successfully.

Use the **set** form of this command to specify the post-login banner.

Use the **delete** form of this command to return to the default post-login banner.

Use the **show** form of this command to view the post-login banner configuration.

system login banner pre-login <banner>

Specifies the pre-login banner.

Syntax

set system login banner pre-login *banner*

delete system login banner pre-login

show system login banner pre-login

Command Mode

Configuration mode.

Configuration Statement

```
system {
   login {
      banner {
         pre-login banner
      }
   }
}
```

Parameters

banner	The banner to be displayed during login after the user enters a login ID. The string must be enclosed in double-quotes. Special characters such as newline (\n) and tab (\t) can also be entered.

Default

The system displays a welcome message.

Usage Guidelines

Use this command to specify the text that will appear when a user enters their login ID.

Use the **set** form of this command to specify the pre-login banner.

Use the **delete** form of this command to return to the default pre-login banner.

Use the **show** form of this command to view the pre-login banner configuration.

system login radius-server <address>

Defines a RADIUS server for user authentication.

Syntax

set system login radius-server *address* [**port** *port* | **secret** *secret* | **timeout** *timeout*]

delete system login radius-server *address* [**port** | **secret** | **timeout**]

show system login radius-server *address* [**port** | **secret** | **timeout**]

Command Mode

Configuration mode.

Configuration Statement

```
system {
    login {
        radius-server address{
            port port
            secret secret
            timeout timeout
        }
    }
}
```

Parameters

address	Multi-node. The IP address of a remote authentication server running the RADIUS protocol. This server can be used to authenticate multiple users. You can define multiple RADIUS servers by creating multiple **radius-server** configuration nodes.
port	Optional. The port to be used for RADIUS traffic. The default is 1812.
secret	The password for the RADIUS server. This must be the same as that recorded on the RADIUS server. Supported characters are alphanumeric and printable special characters (for example, the space character is not permitted). The secret is case-sensitive.

timeout	Optional. The interval, in seconds, after which, if the RADIUS server has not responded, the next configured RADIUS server should be queried. The range is 1 to 30. The default is 2.

Default

None.

Usage Guidelines

Use this command to define a Remote Authentication Dial In User Service (RADIUS) server and specify the information necessary to log on to it.

The RADIUS secret is specified in plain text. RADIUS secrets are stored in plain text on the system, and used as part of a cryptographic operation for transferring authentication information securely over the network. When you view RADIUS secrets, they are displayed in plain text.

NOTE *RADIUS servers are currently not supported in IPv6.*

Use the **set** form of this command to define a RADIUS server.

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

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

system login tacplus-server <address>

Defines a TACACS+ server for user authentication.

Availability

Vyatta Subscription Edition

Syntax

set system login tacplus-server *address* [**port** *port* | **secret** *secret* | **source-address addr** | **timeout** *timeout*]

delete system login tacplus-server *address* [**port** | **secret** | **source-address** | **timeout**]

show system login tacplus-server address [**port** | **secret** | **source-address** | **timeout**]

Command Mode

Configuration mode.

Configuration Statement

```
system {
    login {
        tacplus-server address {
            port port
            secret secret
            source-address addr
            timeout timeout
        }
    }
}
```

Parameters

address	Multi-node. The IP address or hostname of a remote authentication server running TACACS+. This server can be used to authenticate multiple users. You can define multiple TACACS+ servers by creating multiple **tacplus-server** configuration nodes.
port	Optional. The port to be used for TACACS+ traffic. The default is 49.

secret	Mandatory. The password for the TACACS+ server. This must be the same as that recorded on the TACACS+ server. Supported characters are alphanumeric and printable special characters (for example, the space character is not permitted). The secret is case-sensitive.
addr	Optional. The IP address to use as the source address when connecting to the TACACS+ server. This is typically not required.
timeout	Optional. The interval, in seconds, after which, if the TACACS+ server has not responded, the next configured TACACS+ server should be queried. The range is 1 to 30. The default is 3.

Default

None.

Usage Guidelines

Use this command to define a Terminal Access Control Access-Control System Plus (TACACS+) server and specify the information necessary to log on to it.

The TACACS+ secret is specified in plain text. TACACS+ secrets are stored in plain text on the system, and used as part of a cryptographic operation for transferring authentication information securely over the network. When you view TACACS+ secrets, they are displayed in plain text.

NOTE *TACACS+ servers are not supported for IPv6.*

Users doing packet capture and need to see the encrypted TACACS+ traffic

Use the **set** form of this command to define a TACACS+ server.

Use the **delete** form of this command to remove a TACACS+ server.

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

system login user <user>

Creates a user account.

Syntax

set system login user *user*

delete system login user *user*

show system login user *user*

Command Mode

Configuration mode.

Configuration Statement

```
system {
   login {
      user user {}
   }
}
```

Parameters

user	Multi-node. A unique user ID of up to 32 characters, including alphanumeric characters or hyphens. You can define multiple user accounts by creating multiple **user** configuration nodes.

Default

None.

Usage Guidelines

Use this command to define a user that will be authenticated using the system's internal mechanism: "login" authentication.

Note that, although user account and authentication information can be changed using the operating system shell, the system will overwrite these changes the next time you commit configuration in the Vyatta shell. For persistent changes to user or authentication information, use Vyatta CLI commands.

Also, a user cannot be added to the local authentication database if the same username already exists in an accessible remote authentication database (for example, TACACS+).

Use the **set** form of this command to create a **user** configuration node.

Use the **delete** form of this command to remove a **user** configuration node. Note that you cannot delete the account you are currently using.

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

system login user <user> authentication

Sets an authentication password for a user.

Syntax

set system login user user authentication {**encrypted-password** *epwd* | **plaintext-password** *ppwd*}

delete system login user user authentication [**encrypted-password** | **plaintext-password**]

show system login user user authentication [**encrypted-password** | **plaintext-password**]

Command Mode

Configuration mode.

Configuration Statement

```
system {
   login {
      user user {
         authentication {
            encrypted-password epwd
            plaintext-password ppwd
         }
      }
   }
}
```

Parameters

user	The user ID.
epwd	The encrypted password. This value is system generated and should not be altered.
ppwd	The user’s password, specified in plain text. Most special characters can be used with the exceptions of single quote, double quote, and “\”.

Default

None.

Usage Guidelines

Use this command to set a password to authenticate a user. Passwords are automatically encrypted by the system using Message Digest 5 (MD5) encryption. The encrypted version is stored internally and used. When displayed the encrypted value is shown. The plaintext password appears as double quotes in the configuration.

To disable a user account without deleting it, you can simply set the value of the **encrypted-password** option to "*".

Use the **set** form of this command to set a user's password.

Use the **delete** form of this command to remove a user's password.

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

system login user <user> authentication public-keys

Specifies parameters for SSH shared public key user authentication.

Syntax

set system login user *user* **authentication public-keys** *key-id* [**key** *key-value* | **options** *key-options* | **type** *key-type*]

delete system login user *user* **authentication public-keys** *key-id* [**key** | **options** | **type**]

show system login user *user* **authentication public-keys** *key-id* [**key** | **options** | **type**]

Command Mode

Configuration mode.

Configuration Statement

```
system {
    login {
        user user {
            authentication {
                public-keys key-id {
                    key key-value
                    options key-options
                    type key-type
                }
            }
        }
    }
}
```

Parameters

user	The user ID.
key-id	The key identifier. This is typically in the form user@host and is generated by the **ssh-keygen** command when used to create the private and public key pair.
key-value	The shared public key string.

key-options	The optional options which consist of a comma-separated option specification. See the “AUTHORIZED_KEYS FILE FORMAT” section of the **sshd** manual page (**man sshd**) for a detailed description of the available options.
key-type	The authentication type to be used. This parameter must be specified. Supported values are as follows: **ssh-dsa:** Use DSA authentication. **ssh-rsa:** User RSA authentication.

Default

None.

Usage Guidelines

Use this command to specify the parameters to be used for shared public key authentication for logins using SSH. During commit, these values are placed in **/home/<user>/.ssh/authorized_keys**. Changes to this file can only be made using this command. All direct user changes to this file will be lost.

Rather than specifying these parameters directly using the **set** form of this command, the recommended method is to use loadkey command. It will populate the *key-id*, *key-value*, *key-options*, and *key-type* arguments for a specified user given a shared public key file generated by the Linux **ssh-keygen** command on the remote system.

Shared public key authentication for SSH can be available in addition to password authentication for SSH or it can be used exclusively. If both methods are made available at the same time then a login prompt will only appear if a shared public key is not provided at the start of the SSH session. In order to use only shared public keys for SSH authentication, password authentication for SSH must first be disabled. For information on disabling password authentication for SSH see the *Vyatta Services Reference Guide*.

Use the **set** form of this command to set the public key parameters.

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

Use the **show** form of this command to view public key parameters.

system login user <user> full-name <name>

Allows you to record a user’s full name.

Syntax

set system login user user full-name *name*

delete system login user user full-name

show system login user user full-name

Command Mode

Configuration mode.

Configuration Statement

```
system {
    login {
        user user {
            full-name name
        }
    }
}
```

Parameters

user	The user ID.
name	A string representing the user’s name, including alphanumeric characters, space, and hyphens. Strings that include spaces must be enclosed in double quotes.

Default

None.

Usage Guidelines

Use this command to record a user’s full name.

Use the **set** form of this command to specify the user’s name.

Use the **delete** form of this command to remove the user’s name.

Use the **show** form of this command to view a user’s name.

system login user <user> group <group>

Allows you to make a user a member of a group.

Syntax

set system login user *user* **group** *group*

delete system login user *user* **group**

show system login user *user* **group**

Command Mode

Configuration mode.

Configuration Statement

```
system {
   login {
      user user {
         group group
      }
   }
}
```

Parameters

user	The user ID.
group	A string representing the the group the user is to be assigned to. Groups are defined in the **/etc/group** directory.

Default

None

Usage Guidelines

Use this command to assign a user to a group. Users can be members of multiple groups by executing this command once for each group the user is to be assigned to.

Use the **set** form of this command to make a user a member of the specified group.

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

Use the **show** form of this command to view the groups that the user is assigned to.

system login user <user> home-directory <dir>

Allows you to specify a user’s home directory.

Syntax

set system login user user home-directory *dir*

delete system login user user home-directory

show system login user user home-directory

Command Mode

Configuration mode.

Configuration Statement

```
system {
   login {
      user user {
         home-directory dir
      }
   }
}
```

Parameters

user	The user ID.
dir	A string representing the user’s home directory; for example, **/home/vyatta**.

Default

The home directory is **/home/<user>**.

Usage Guidelines

Use this command to specify a user’s home directory.

Use the **set** form of this command to specify the user’s home directory.

Use the **delete** form of this command to restore the user’s default home directory.

Use the **show** form of this command to view a user’s home directory.

system login user <user> level <level>

Specifies a user’s privilege level and system access.

Syntax

set system login user *user* **level** *level*

delete system login user *user* **level**

show system login user *user* **level**

Command Mode

Configuration mode.

Configuration Statement

```
system {
    login {
        user user {
            level level
        }
    }
}
```

Parameters

user	The user ID.
level	Determines the user’s level of priviledge. Supported values are as follows: **admin:** Assigns the user administrative privileges. The user can execute any command in the Vyatta CLI or the underlying operating system. **operator:** Assigns the user restricted privileges. The user can execute operational commands in the Vyatta CLI, plus a restricted form of **ping** and **traceroute**. The user cannot enter configuration mode or execute configuration commands.

Default

Users are assigned administrative privileges by default.

Usage Guidelines

Use this command to assign role-based system access to a user.

The system supports two system roles:

- Admin user. Users assigned a role of admin have full access to all Vyatta-specific commands plus all operating system shell commands. Access to operating system shell commands is direct: the user need not exit to another shell mode before executing these commands. Although admin users can execute any command implemented in the system, command completion and CLI help show only Vyatta commands.
- Operator user. Users assigned a role of operator have access to the Vyatta operational command set, but no access to configuration commands. They also have limited access to operating system commands. At this time, command completion and CLI help show all Vyatta commands for users with the operator role.

Use the **set** form of this command to set a user’s privilege level.

Use the **delete** form of this command to restore a user’s privilege level to the default.

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

system tacplus-options command-accounting

Enables logging accounting records for interactive shell commands.

Syntax

set system tacplus-options command-accounting

delete system tacplus-options command-accounting

show system tacplus-options

Command Mode

Configuration mode.

Configuration Statement

```
system {
   tacplus-options {
      command-accounting
   }
}
```

Parameters

None.

Default

Accounting records are not logged.

Usage Guidelines

Use this command to enable logging accounting records for interactive shell (**vbash**) commands.

Connections for which commands are logged include SSH, Telnet, console, and serial connections to the system. Command logging is not limited to TACACS+ authenticated users and accounts for **vbash** shell commands. Accounting records are logged to the TACACS+ server.

Use the **set** form of this command to enable logging accounting records for interactive shell commands.

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

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

Chapter 5: Flow Accounting

This chapter explains how to configure flow accounting using the Vyatta system.

This chapter presents the following topics:

Flow Accounting Configuration

This section presents the following topics:

- Flow Accounting Overview
- Configuring an Interface for Flow Accounting
- Displaying Flow Accounting Information
- Exporting Flow Accounting information

Flow Accounting Overview

Flow accounting provides the ability to locally display information about network traffic, as well as the ability to export this information to Netflow- or sFlow-compatible collection servers.

A network flow is defined as a unidirectional sequence of packets all of which have a common source IP address, destination IP address, source port (for UDP or TCP, 0 for other protocols), destination port (for UDP or TCP, type and code for ICMP, 0 for other protocols), IP protocol, ingress interface, and Type of Service.

Each separate TCP session with identical network flow information is counted as a new flow in the statistics. A TCP flow is considered complete if its session completes or the flow times out. There are a number of available timeout values that can be configured, as required.

For connectionless protocols like ICMP and UDP, a flow is considered complete after no packets for that flow appear for a configurable timeout period.

Flow accounting is defined on a per-interface basis. All packets recieved by the interface can be counted, resulting in very precise statistics. However, viewing all packets consumes significant computing resources. An alternative is to sample every *n* packets (the sampling rate) and to estimate data traffic based on these samples. This consumes fewer system resources than viewing all packets, especially for large data volumes, while still providing reasonable accuracy.

Configuring an Interface for Flow Accounting

In order for flow accounting information to be collected and displayed for an interface, the interface must first be configured for flow accounting. The following example shows how to configure eth0 for flow accounting in configuration mode.

Example 5-1 Configuring an interface for flow accounting

Step	Command
Configure flow accounting on eth0.	`vyatta@vyatta#` **`set system flow-accounting interface eth0`**

Example 5-1 Configuring an interface for flow accounting

Step	Command
Commit the configuration.	vyatta@vyatta# **commit**

Displaying Flow Accounting Information

Once flow accounting is configured on selected interfaces it provides the ability to display network traffic information for all configured interfaces, by interface, by interface and host, by interface and port, as well as by traffic volume on an interface. The following operational mode example shows flow accounting for eth0.

Example 5-2 Showing flow accounting information for eth0

```
vyatta@vyatta:~$ show flow-accounting interface eth0
flow-accounting for [eth0]
Src Addr        Dst Addr        Sport Dport Proto    Packets      Bytes   Flows
192.168.1.156   192.168.1.80    3024  22      tcp         98       6520       0
192.168.1.8     255.255.255.255 22936 2220    udp          2        696       1
192.168.1.8     255.255.255.255 22936 3245    udp          2        696       1
192.168.1.8     255.255.255.255 22936 2214    udp          2        696       1
192.168.1.8     255.255.255.255 22936 3242    udp          2        696       1
192.168.1.156   192.168.1.255   138   138     udp          2        480       1
192.168.1.8     192.168.1.255   138   138     udp          1        240       1
192.168.1.10    192.168.1.255   2214  22936   udp          4        240       1
192.168.1.156   192.168.1.255   3245  22936   udp          4        240       1
192.168.1.10    192.168.1.255   2220  22936   udp          4        240       1
192.168.1.156   192.168.1.255   3242  22936   udp          4        240       1
192.168.1.8     192.168.1.255   137   137     udp          1         78       1

Total entries: 12
Total flows  : 11
Total pkts   : 126
Total bytes  : 11,062
vyatta@vyatta:~$
```

The following example shows flow accounting for host 192.168.1.156 on eth0.

Example 5-3 Showing flow accounting information for 192.168.1.156 on eth0

```
vyatta@vyatta:~$ show flow-accounting interface eth0 host 192.168.1.156
Src Addr        Dst Addr        Sport Dport Proto    Packets      Bytes   Flows
192.168.1.156   192.168.1.80    3024  22      tcp        107       7036       0
192.168.1.156   192.168.1.255   138   138     udp          2        480       1
192.168.1.156   192.168.1.255   3245  22936   udp          4        240       1
```

```
192.168.1.156   192.168.1.255   3242  22936   udp          4        240       1

Total entries: 4
Total flows  : 3
Total pkts   : 117
Total bytes  : 7,996
vyatta@vyatta:~$
```

Exporting Flow Accounting information

In addition to displaying flow accounting information locally, this information can be exported to a collection server. The following example shows how to configure the system to export flow accounting information in Netflow format to a collection server with IP address 192.168.1.20 on the default port.

Example 5-4 Exporting data in Netflow format to 192.168.1.20

Step	Command
Configure the export of data in Netflow format to 192.168.1.20.	`vyatta@vyatta#` **`set system flow-accounting netflow server 192.168.1.20`**
Commit the configuration.	`vyatta@vyatta#` **`commit`**

Flow Accounting Commands

This section presents the following commands.

Configuration Commands	
system flow-accounting interface <interface>	Specifies the interface on which to record inbound flow statistics.
system flow-accounting netflow engine-id <id>	Specifies the system ID to appear in Netflow data.
system flow-accounting netflow sampling-rate <rate>	Specifies the rate at which packets are sampled for statistics.
system flow-accounting netflow server <ipv4>	Specifies a Netflow collector to which to export Netflow data.
system flow-accounting netflow timeout expiry-interval <interval>	Specifies the interval at which Netflow data will be sent to a Netflow collector.
system flow-accounting netflow timeout flow-generic <timeout>	Specifies the flow timeout for generic IP traffic.
system flow-accounting netflow timeout icmp <timeout>	Specifies the flow timeout for ICMP traffic.
system flow-accounting netflow timeout max-active-life <life>	Specifies the maximum time for which any flow can have data collected.
system flow-accounting netflow timeout tcp-fin <timeout>	Specifies the TCP flow timeout after receiving a TCP FIN packet.
system flow-accounting netflow timeout tcp-generic <timeout>	Specifies the generic TCP flow timeout.
system flow-accounting netflow timeout tcp-rst <timeout>	Specifies the TCP flow timeout after receiving a TCP RST packet.
system flow-accounting netflow timeout udp <timeout>	Specifies the flow timeout for UDP traffic.
system flow-accounting netflow version <version>	Specifies the Netflow format that data will be exported in.
system flow-accounting sflow agent-address <addr>	Allows you to spcify the IP address of the sFlow agent.
system flow-accounting sflow sampling-rate <rate>	Specifies the rate at which sFlow statistics are recorded.
system flow-accounting sflow server <ipv4>	Specifies an sflow collector to export sFlow data to.

Configuration Commands	
system flow-accounting syslog-facility <facility>	Specifies the kinds of flow accounting messages to be logged.

Operational Commands	
clear flow-accounting counters	Clears all flow accounting counters.
restart flow-accounting	Restarts the flow accounting process.
show flow-accounting	Displays flow statistics for all interfaces on which flow accounting is enabled.
show flow-accounting interface <interface>	Displays flow statistics for a specific interface configured for flow accounting.

clear flow-accounting counters

Clears all flow accounting counters.

Syntax

clear flow-accounting counters

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to clear flow accounting counters on all configured interfaces.

restart flow-accounting

Restarts the flow accounting process.

Syntax

restart flow-accounting process

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to restart the flow accounting process.

show flow-accounting

Displays flow statistics for all interfaces on which flow accounting is enabled.

Syntax

show flow-accounting

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display flow statistics for all interfaces configured for flow accounting. Statistics are displayed for each interface configured for flow accounting.

show flow-accounting interface <interface>

Displays flow statistics for a specific interface configured for flow accounting.

Syntax

show flow-accounting interface *interface* [**host** *host*] [**port** *port*] [**top** *top*]

Command Mode

Operational mode.

Parameters

interface	The interface from which to obtain flow statistics (for example, eth0). This interface must first be configured for flow accounting.
host	The IP address of a specific host whose flow statistics are to be displayed.
port	The port number of a specific port whose flow statistics are to be displayed.
top	The number of flows with the heaviest traffic to be displayed. They are displayed in decending order based on the number of bytes received on the interface.

Default

None.

Usage Guidelines

Use this command to display flow statistics for the specified interface. The interface must first be configured for flow accounting.

system flow-accounting interface <interface>

Specifies the interface on which to record inbound flow statistics.

Syntax

set system flow-accounting interface *interface*

delete system flow-accounting interface *interface*

show system flow-accounting *interface*

Command Mode

Configuration mode.

Configuration Statement

```
system {
   flow-accounting {
      interface interface
   }
}
```

Parameters

interface	Multi-node. The interface on which to record inbound flow statistics (for example, eth0). You can enable multiple interfaces for flow accounting by creating multiple **interface** configuration nodes.

Default

None.

Usage Guidelines

Use this command to configure an interface to record inbound flow statistics.

Use the **set** form of this command to configure an interface to record inbound flow statistics.

Use the **delete** form of this command to stop an interface from recording inbound flow statistics.

Use the **show** form of this command to show the interfaces configured to record inbound flow statistics.

system flow-accounting netflow engine-id <id>

Specifies the system ID to appear in Netflow data.

Syntax

set system flow-accounting netflow engine-id *id*

delete system flow-accounting netflow engine-id

show system flow-accounting netflow engine-id

Command Mode

Configuration mode.

Configuration Statement

```
system {
    flow-accounting {
        netflow {
            engine-id id
        }
    }
}
```

Parameters

id	The system ID that will appear in Netflow data indentifying the router that the data came from. The range is 0 to 255.

Default

None.

Usage Guidelines

Use this command to configure the system ID to appear in Netflow data.

Use the **set** form of this command to configure the system ID to appear in Netflow data.

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

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

system flow-accounting netflow sampling-rate <rate>

Specifies the rate at which packets are sampled for statistics.

Syntax

set system flow-accounting netflow sampling-rate *rate*

delete system flow-accounting netflow sampling-rate

show system flow-accounting netflow sampling-rate

Command Mode

Configuration mode.

Configuration Statement

```
system {
   flow-accounting {
      netflow {
         sampling-rate rate
      }
   }
}
```

Parameters

rate	The rate at which packets are sampled (that is, if 1 in *n* packets are sampled, *n* is the rate).

Default

Every packet is sampled (that is, the sampling rate is 1).

Usage Guidelines

Use this command to configure the Netflow sampling rate for flow accounting. The system samples one in every *n* packets, where *n* is the value configured for the **sampling-rate** option.

The advantage of sampling every *n* packets, where *n* > 1, allows you to decrease the amount of processing resources required for flow accounting. The disadvantage of not sampling every packet is that the statistics produced are estimates of actual data flows.

Use the **set** form of this command to specfy the sampling rate.

Use the **delete** form of this command to sample all packets.

Use the **show** form of this command to display sampling rate configuration.

system flow-accounting netflow server <ipv4>

Specifies a Netflow collector to which to export Netflow data.

Syntax

set system flow-accounting netflow server *ipv4* [**port** *port*]

delete system flow-accounting netflow server *ipv4* [**port**]

show system flow-accounting netflow server *ipv4* [**port**]

Command Mode

Configuration mode.

Configuration Statement

```
system {
   flow-accounting {
      netflow {
         server ipv4 {
            port port
         }
      }
   }
}
```

Parameters

ipv4	Multi-node. The IP address of a Netflow collector to which to export the Netflow data. You can export Netflow data to more than collector by issuing this command multiple times.
port	The port on the Netflow collector to which to export the Netflow. The default value is 2055.

Default

None.

Usage Guidelines

Use this command to specify a Netflow collector for exporting flow accounting data.

Use the **set** form of this command to specify a Netflow collector.

Use the **delete** form of this command to remove a Netflow collector configuration.

Use the **show** form of this command to display Netflow collector configuration.

system flow-accounting netflow timeout expiry-interval <interval>

Specifies the interval at which Netflow data will be sent to a Netflow collector.

Syntax

set system flow-accounting netflow timeout expiry-interval *interval*

delete system flow-accounting netflow timeout expiry-interval

show system flow-accounting netflow timeout expiry-interval

Command Mode

Configuration mode.

Configuration Statement

```
system {
   flow-accounting {
      netflow {
         timeout {
            expiry-interval interval
         }
      }
   }
}
```

Parameters

interval	The interval at which Netflow data will be sent to a Netflow collector.

Default

Netflow data will be sent every 60 seconds.

Usage Guidelines

Use this command to configure the interval at which the system will send Netflow data to a Neflow collector. The Netflow collector must first be configured using the system flow-accounting netflow server <ipv4> command.

Use the **set** form of this command to configure the interval at which the system will send Netflow data to a Neflow collector.

Use the **delete** form of this command to return the system to the default value interval.

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

system flow-accounting netflow timeout flow-generic <timeout>

Specifies the flow timeout for generic IP traffic.

Syntax

set system flow-accounting netflow timeout flow-generic *timeout*

delete system flow-accounting netflow timeout flow-generic

show system flow-accounting netflow timeout flow-generic

Command Mode

Configuration mode.

Configuration Statement

```
system {
    flow-accounting {
        netflow {
            timeout {
                flow-generic timeout
            }
        }
    }
}
```

Parameters

timeout	The flow timeout, in seconds, for generic IP traffic. This includes all IP traffic except TCP, UDP, and ICMP. The range is 1 to 4294967296. The default value is 3600 (1 hour).

Default

Generic IP traffic flows time out after 3600 seconds.

Usage Guidelines

Use this command to configure the flow timeout for generic IP traffic. Generic IP traffic consists of all IP traffic except TCP, UDP, and ICMP. (Generic IP traffic would include, for example, GRE, AH, ESP, and so on.)

This parameter defines the amount of time the system continues to wait for data from a generic IP flow before considering the flow complete.

Use the **set** form of this command to set the flow timeout for generic IP traffic.

Use the **delete** form of this command to return the flow timeout for generic IP traffic to the default value.

Use the **show** form of this command to view generic IP traffic flow timeout configuration.

system flow-accounting netflow timeout icmp <timeout>

Specifies the flow timeout for ICMP traffic.

Syntax

set system flow-accounting netflow timeout icmp *timeout*

delete system flow-accounting netflow timeout icmp

show system flow-accounting netflow timeout icmp

Command Mode

Configuration mode.

Configuration Statement

```
system {
   flow-accounting {
      netflow {
         timeout {
            icmp timeout
         }
      }
   }
}
```

Parameters

timeout	The flow timeout, in seconds, for ICMP traffic. The range is 1 to 4294967296. The default value is 300 (5 minutes).

Default

ICMP traffic flows timeout after 300 seconds.

Usage Guidelines

Use this command to configure the flow timeout for ICMP traffic. This parameter defines the amount of time the system continues to wait for data from an ICMP flow before considering the flow complete.

Use the **set** form of this command to set the flow timeout for ICMP traffic.

Use the **delete** form of this command to to return the flow timeout for ICMP traffic to the default value.

Use the **show** form of this command to view ICMP traffic flow timeout configuration.

system flow-accounting netflow timeout max-active-life <life>

Specifies the maximum time for which any flow can have data collected.

Syntax

set system flow-accounting netflow timeout max-active-life *life*

delete system flow-accounting netflow timeout max-active-life

show system flow-accounting netflow timeout max-active-life

Command Mode

Configuration mode.

Configuration Statement

```
system {
    flow-accounting {
        netflow {
            timeout {
                max-active-life life
            }
        }
    }
}
```

Parameters

life	The global flow timeout, in seconds. The range is 1 to 4294967296. The default value is 604800 (7 days).

Default

All flows time out after 604,800 seconds.

Usage Guidelines

Use this command to configure the global flow timeout.

This parameter defines the amount of time the system continues to wait for data from any flow before considering the flow complete. Even if the flow is still active when it reaches this timeout value, it will be considered complete from a flow accounting perspective.

Use the **set** form of this command to set the global flow timeout.

Use the **delete** form of this command to to return the global flow timeout to the default value.

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

system flow-accounting netflow timeout tcp-fin <timeout>

Specifies the TCP flow timeout after receiving a TCP FIN packet.

Syntax

set system flow-accounting netflow timeout tcp-fin *timeout*

delete system flow-accounting netflow timeout tcp-fin

show system flow-accounting netflow timeout tcp-fin

Command Mode

Configuration mode.

Configuration Statement

```
system {
    flow-accounting {
        netflow {
            timeout {
                tcp-fin timeout
            }
        }
    }
}
```

Parameters

timeout	The flow timeout, in seconds, after receiving a TCP FIN packet. The range is 1 to 4294967296. The default value is 300 (5 minutes).

Default

A TCP flow times out 300 seconds after receiving a TCP FIN packet without receiving the corresponding FIN ACK, ACK sequence.

Usage Guidelines

Use this command to configure the TCP flow timeout after receiving a TCP FIN packet. This parameter defines the amount of time the system continues to wait for data from a TCP flow after receiving a TCP FIN packet without having received the corresponding FIN ACK, ACK sequence. When this timeout expires, the flow is considered complete.

Use the **set** form of this command to set the TCP FIN flow timeout.

Use the **delete** form of this command to return the TCP FIN flow timeout to the default value.

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

system flow-accounting netflow timeout tcp-generic <timeout>

Specifies the generic TCP flow timeout.

Syntax

set system flow-accounting netflow timeout tcp-generic *timeout*

delete system flow-accounting netflow timeout tcp-generic

show system flow-accounting netflow timeout tcp-generic

Command Mode

Configuration mode.

Configuration Statement

```
system {
    flow-accounting {
        netflow {
            timeout {
                tcp-generic timeout
            }
        }
    }
}
```

Parameters

timeout	The generic TCP flow timeout, in seconds. The range is 1 to 4294967296. The default value is 3600 (1 hour).

Default

A TCP flow will timeout 3600 seconds after seeing no data or TCP FIN, FIN ACK, ACK sequence.

Usage Guidelines

Use this command to configure the TCP flow timeout after seeing no data or TCP FIN, FIN ACK, ACK sequence. This parameter defines the amount of time the system will continue to wait for data from a TCP flow without seeing any data, or a TCP FIN, and the corresponding FIN ACK, ACK sequence, before considering the flow complete.

Use the **set** form of this command to set the generic TCP flow timeout.

Use the **delete** form of this command to return the generic TCP flow timeout to the default value.

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

system flow-accounting netflow timeout tcp-rst <timeout>

Specifies the TCP flow timeout after receiving a TCP RST packet.

Syntax

set system flow-accounting netflow timeout tcp-rst *timeout*

delete system flow-accounting netflow timeout tcp-rst

show system flow-accounting netflow timeout tcp-rst

Command Mode

Configuration mode.

Configuration Statement

```
system {
   flow-accounting {
      netflow {
         timeout {
            tcp-rst timeout
         }
      }
   }
}
```

Parameters

timeout	The flow timeout after receiving a TCP RST packet. The range is 1 to 4294967296. The default value is 120 (2 minutes).

Default

A TCP flow will timeout 120 seconds after seeing a TCP RST packet without seeing any other packets (i.e. data, TCP FIN, FIN ACK, or ACK).

Usage Guidelines

Use this command to configure the TCP flow timeout after seeing a TCP RST packet but no data,TCP FIN, FIN ACK, or ACK. This parameter defines the amount of time the system will continue to wait for data from a TCP flow after seeing a TSCP RST but without seeing any data, TCP FIN, FIN ACK, or ACK packets, before considering the flow complete.

Use the **set** form of this command to set the TCP RST flow timeout.

Use the **delete** form of this command to return the TCP RST flow timeout to the default value.

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

system flow-accounting netflow timeout udp <timeout>

Specifies the flow timeout for UDP traffic.

Syntax

set system flow-accounting netflow timeout udp *timeout*

delete system flow-accounting netflow timeout udp

show system flow-accounting netflow timeout udp

Command Mode

Configuration mode.

Configuration Statement

```
system {
   flow-accounting {
      netflow {
         timeout {
            udp timeout
         }
      }
   }
}
```

Parameters

timeout	The flow timeout for UDP traffic.The range is 1 to 4294967296. The default value is 300 (5 minutes).

Default

UDP traffic flows timeout after 300 seconds.

Usage Guidelines

Use this command to configure the flow timeout for UDP traffic. This parameter defines the amount of time the system will continue to wait for data from an UDP flow before considering the flow complete.

Use the **set** form of this command to set the flow timeout for UDP traffic.

Use the **delete** form of this command to to return the flow timeout for UDP traffic to the default value.

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

system flow-accounting netflow version <version>

Specifies the Netflow format that data will be exported in.

Syntax

set system flow-accounting netflow version *version*

delete system flow-accounting netflow version

show system flow-accounting netflow version

Command Mode

Configuration mode.

Configuration Statement

```
system {
    flow-accounting {
        netflow {
            version version
        }
    }
}
```

Parameters

version	The Netflow version the exported data is formatted in. Supported values are 1, 5, and 9. The default value is 5.

Default

Netflow version 5 format is used.

Usage Guidelines

Use this command to set the formatting of the exported data to match a Netflow version.

Use the **set** form of this command to specify the Netflow version.

Use the **delete** form of this command to remove the configured version number and use the default value.

Use the **show** form of this command to display Netflow version configuration.

system flow-accounting sflow agent-address <addr>

Allows you to spcify the IP address of the sFlow agent.

Syntax

set system flow-accounting sflow agent-address *addr*

delete system flow-accounting sflow agent-address

show system flow-accounting sflow agent-address

Command Mode

Configuration mode.

Configuration Statement

```
system {
    flow-accounting {
        sflow {
            agent-address addr
        }
    }
}
```

Parameters

addr	The IP address of the sFlow agent to be included in sFlow packets sent to the collector. Supported values are **auto** (in which case, the system selects one of its own IP address) or an IPv4 address. The default value is **auto**.

Default

The system selects an IP address to send as the source for sFlow data.

Usage Guidelines

Use this command to configure an IP address to be sent to the sFlow collector indicating the source of the sFlow data—i.e., the local Vyatta system.

Use the **set** form of this command to set the agent address.

Use the **delete** form of this command to remove the agent address and use the default.

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

system flow-accounting sflow sampling-rate <rate>

Specifies the rate at which sFlow statistics are recorded.

Syntax

set system flow-accounting sflow sampling-rate *rate*

delete system flow-accounting sflow sampling-rate

show system flow-accounting sflow sampling-rate

Command Mode

Configuration mode.

Configuration Statement

```
system {
    flow-accounting {
        sflow {
            sampling-rate rate
        }
    }
}
```

Parameters

rate	The rate at which packets are sampled (that is, if 1 in *n* packets are sampled, *n* is the rate).

Default

Every packet is sampled (that is, the sampling rate is 1).

Usage Guidelines

Use this command to configure the sFlow sampling rate for flow accounting. The system samples one in every *n* packets, where *n* is the value configured for the **sampling-rate** option.

The advantage of sampling every *n* packets, where *n* > 1, allows you to decrease the amount of processing resources required for flow accounting. The disadvantage of not sampling every packet is that the statistics produced are estimates of actual data flows.

Use the **set** form of this command to specfy the sampling rate.

Use the **delete** form of this command to sample all packets.

Use the **show** form of this command to display sampling rate configuration.

system flow-accounting sflow server <ipv4>

Specifies an sflow collector to export sFlow data to.

Syntax

set system flow-accounting sflow server *ipv4* [**port** *port*]

delete system flow-accounting sflow server *ipv4* [**port**]

show system flow-accounting sflow server *ipv4* [**port**]

Command Mode

Configuration mode.

Configuration Statement

```
system {
    flow-accounting {
        sflow {
            server ipv4 {
                port port
            }
        }
    }
}
```

Parameters

ipv4	Multi-node. The IP address of an sFlow collector to export the sFlow data to. You can export sFlow data to more than one sFlow collector by issuing this command multiple times.
port	The port on the sFlow collector to export the sFlow data to. The default value is 6343.

Default

None.

Usage Guidelines

Use this command to specify an sFlow collector to which to export sFlow data.

Use the **set** form of this command to specify an sFlow collector.

Use the **delete** form of this command to remove an sFlow collector configuration.

Use the **show** form of this command to display sFlow collector configuration.

system flow-accounting syslog-facility <facility>

Specifies the kinds of flow accounting messages to be logged.

Syntax

set system flow-accounting syslog-facility *facility*

delete system flow-accounting syslog-facility

show system flow-accounting syslog-facility

Command Mode

Configuration mode.

Configuration Statement

```
system {
   flow-accounting {
      syslog-facility facility
   }
}
```

Parameters

facility	The kinds of messages to be logged using **syslog**. Please see the Usage Guidelines in the **system syslog** command for supported facilities. The default value is **daemon**.

Default

System daemon messages are logged.

Usage Guidelines

Use this command to configure the kinds of flow accounting messages that will be logged.

Use the **set** form of this command to specify the kinds of flow accounting messages that will be logged.

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

Use the **show** form of this command to display configuration flow accounting logging configuration.

Chapter 6: Connection Tracking

This chapter explains connection tracking in the Vyatta system.

This chapter presents the following topics:

- Connection Tracking Overview
- Connection Tracking Commands

Connection Tracking Overview

The Vyatta system keeps track of connections for some system components—firewall, NAT, WAN load balancing, and web proxy. The connection tracking system consists of four component:

- The connection tracking table
- The connection tracking hash table
- The connection tracking expect table
- The connection tracking expect hash table.

The connection tracking table contains one entry for each connection being tracked by the system. Each entry is approximately 300 bytes and is dynamically allocated as required. The table has a maximum of 16,384 entries if the firewall is not enabled, and 32,768 entries if the firewall is enabled. This value can be changed using the **firewall conntrack-table-size** command in order to tune the system.

The connection tracking hash table makes searching the connection tracking table faster. The hash table uses "buckets" to record entries in the connection tracking table. By default, there are 4096 buckets in the table and each is 8 bytes. Memory for the connection tracking hash table is statically allocated. The size of the connection tracking hash table can be tuned using the **firewall conntrack-hash-size** command. The larger the hash table size, the more static memory is used but the faster the lookup time. The smaller the hash table, size the lower the static memory usage but the slower the lookup time.Typically, the connection tracking hash table is kept at one-eight the number of entries in the connection tracking table.

The connection tracking expect table contains one entry for each expected connection related to an existing connection. These are generally used by "connection tracking helpers" (sometimes called "application-level gateways") for protocols such as FTP, SIP, and H.323. Each entry is approximately 300 bytes and is dynamically allocated as required, up to a maximum of 2048 entries if the firewall is not enabled, and 4096 entries if the firewall is enabled. This value can be tuned using the **firewall conntrack-expect-table-size** command.

The connection tracking expect hash table is used to make searching the connection tracking expect table faster There are 1024 eight-byte buckets in the table. Memory for the connection tracking expect hash table is statically allocated. The size of the connection tracking expect hash table is not currently configurable using the Vyatta CLI.

For most installations, the default values of these tables will serve well. For high capacity systems where the number of simultaneous connections is potentially more than the connection tracking table can hold, the table sizes can be increased, keeping in mind the following:

1. Each entry in the connection tracking table and the connection tracking expect table is approximately 300 bytes. This memory is dynamically allocated as required.

2. Each bucket in the connection tracking hash table is eight bytes. This memory is statically allocated.
3. For reasonable lookup speed, there should be approximately one bucket in the connection tracking hash table for every eight entries in the connection tracking table.
4. For better look-up performance, increase the size of the connection tracking hash table with respect to the connection tracking table. It doesn’t make sense to bring the ratio for the size of these two tables closer than 1:1 (for example, if the connection tracking table is set to 65,536 then the maximum hash table size should not be greater than 65,536 as well).
5. Calculate the mamximum amount of memory that could be used by the connection tracking tables and make sure that you have enough free physical memory to accomodate them at their maximum size.

Connection Tracking Commands

This section presents the following commands.

Operational Commands	
clear connection-tracking	Flushes the connection tracking table.

Configuration commands that manipulate the connection tracking–related tables are located in the *Vyatta Firewall Reference Guide*.

clear connection-tracking

Flushes the connection tracking table.

Syntax

clear connection-tracking

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to flush all connections currently being tracked in the connection tracking table.

Chapter 7: Logging

This chapter describes the Vyatta system logging mechanism.

This chapter presents the following topics:

- Logging Configuration
- Logging Commands

Logging Configuration

This section presents the following topics:

- Logging Overview
- Logging Configuration Example
- Enabling and Disabling Logging for Specific Features

Logging Overview

Significant system events are captured in log messages (also called syslog messages), which you can view on the console, save to a file, or forward to an external server such as a syslog server, or direct to the terminal session of one or more specific users.

Depending on the level of message severity you choose to log, system log messages can include notices of ordinary and routine operations, as well as warnings, failure, and error messages.

The Vyatta System's logging function makes use of the UNIX **syslogd** process. Logging configuration performed within the system's CLI is stored in the **/etc/sylogd.conf** file.

By default, local logging is enabled, and sends messages to **/var/log/messages**.

Logging Facilities

The Vyatta system supports standard syslog facilities. These are as follows:

Table 7-1 Syslog facilities

Facility	Description
auth	Authentication and authorization
authpriv	Non-system authorization
cron	Cron daemon
daemon	System daemons
kern	Kernel
lpr	Line printer spooler
mail	Mail subsystem
mark	Timestamp
news	USENET subsystem

Table 7-1 Syslog facilities

security	Security subsystem
syslog	System logging
user	Application processes
uucp	UUCP subsystem
local0	Local facility 0
local1	Local facility 1
local2	Local facility 2
local3	Local facility 3
local4	Local facility 4
local5	Local facility 5
local6	Local facility 6
local7	Local facility 7
all	All facilities excluding “mark”

In addition, logging can be selectively enabled for some specific routing components. For this information, please see the section ““Enabling and Disabling Logging for Specific Features” on page 283.

Log Destinations

When logging is enabled, system log messages are always written to the **messages** file in the **/var/log** directory of the local file system. In addition, system logs can be sent to the console, to a named file in the local file system, to a server running the **syslogd** utility (that is, a syslog server), or to the terminal session of one or more specific users.

- To direct syslog messages to the console, use the **system syslog console** command.
- To direct syslog messages to a named file in the local file system, use the **system syslog file** command.
- To direct syslog messages to a remote machine running the **syslogd** utility, use the **system syslog host** command.
- To direct syslog messages to the terminal of a specific user, to multiple users, or to all users logged into the routing platform, use the **system syslog user** command.

Log File Locations and Archiving

Messages are written either to the main log file (the default) or to a file that you specify. User-defined log files are written to the **/var/log/user** directory, under the user-specified file name.

The system uses standard UNIX log rotation to prevent the file system from filling up with log files. When log messages are written to a file, the system will write up to 500 KB of log messages into the file *logfile*, where *logfile* is either the main log file or a name you have assigned to a user-defined file. When *logfile* reaches its maximum size, the system closes it and compresses it into an archive file. The archive file is named *logfile***.0.gz**.

At this point, the logging utility opens a new *logfile* file and begins to write system messages to it. When the new log file is full, the first archive file is renamed *logfile***.1.gz** and the new archive file is named *logfile***.0.gz**.

The system archives log files in this way until a maximum number of log files exists. By default, the maximum number of archived files is 10 (that is, up to *logfile***.9.gz**), where *logfile***.0.gz** always represents the most recent file. After this, the oldest log archive file is deleted as it is overwritten by the next oldest file.

To change the properties of log file archiving, configure the **system syslog archive** node:

- Use the **size** parameter to specify the maximum size of each archived log file.
- Use the **files** parameter to specify the maximum number of archive files to be maintained.

Log Severities

Log messages generated by the Vyatta system will be associated with one of the following levels of severity.

Table 7-2 Syslog message severities

Severity	Meaning
emerg	Emergency. A general system failure or other serious failure has occurred, such that the system is unusable.
alert	Alert. Immediate action is required to prevent the system from becoming unusable—for example, because a network link has failed, or the database has become compromised.
crit	Critical. A critical condition exists, such as resource exhaustion—for example, the system is out of memory, CPU processing thresholds are being exceeded, or a hardware failure has occurred.
err	Error. An error condition has occurred, such as a failed system call. However, the system is still functioning.

Table 7-2 Syslog message severities

warning	Warning. An event has occurred that has the potential to cause an error, such as invalid parameters being passed to a function. This situation should be monitored.
notice	Notice. A normal but significant event has occurred, such as an unexpected event. It is not an error, but could potentially require attention.
info	Informational. Normal events of interest are being reported as they occur.
debug	Debug level. Trace-level information is being provided.

CAUTION *Risk of service degradation. Debug severity is resource-intensive. Setting logging levels to Debug can affect performance.*

Logging Configuration Example

Example 7-1 creates a log file that captures kernel-related alerts of critical and higher severity.

To create a log file to capture kernel-related critical alerts, perform the following steps in configuration mode:

Example 7-1 Configuring a log to capture kernel-related alerts of critical and higher severity

Step	Command
Create a logfile called “kernel-log” and log kernel-related messages of “critical” and higher severity.	`vyatta@R1# set system syslog file kernel-log facility kern level crit`

Commit the configuration.	`vyatta@R1# commit`
	`Restarting system log daemon....`
	`vyatta@R1#`

The command “**show log file** *kernel-log*” can then be used in opertional mode to display the contents of the *kernel-log* logfile.

Enabling and Disabling Logging for Specific Features

Some features of the Vyatta router—for example, BGP, OSPF, and IPsec VPN—produce feature-specific log messages that can be enabled and disabled within the configuration node for that feature. When you enable logging for a system feature, the log messages are sent to whatever destinations are configured for syslog.

By default, log messages are sent to the main log file. You can configure syslog to send log messages to a file you specify in **/var/user**.

Logging Commands

This section presents the following commands.

Configuration Commands	
system syslog	Configures the system's syslog utility.
system syslog console facility <facility> level <level>	Specifies which messages are sent to the console.
system syslog file <filename> archive	Specifies the settings for log file archiving of the user-defined log file.
system syslog file <filename> facility <facility> level <level>	Specifies which messages are sent to the user-defined log file.
system syslog global archive	Specifies the settings for log file archiving of the main system log file.
system syslog global facility <facility> level <level>	Specifies which messages are sent to the main system log file.
system syslog host <hostname> facility <facility> level <level>	Specifies which messages are sent to the remote syslog server.
system syslog user <userid> facility <facility> level <level>	Specifies which messages are sent to the specified user's terminal.
Operational Commands	
delete log file	Deletes the specified log file, including all its archive files.
show log	Displays the contents of the specified log file.
show log image <image-name>	Displays the contents of the log file in the specified image.

delete log file

Deletes the specified log file, including all its archive files.

Syntax

delete log file *file-name*

Command Mode

Operational mode.

Parameters

file-name	Deletes the specified user-defined file in the **/var/log** directory, including all its archive files.

Usage Guidelines

Use this command to delete a log file.

Log files are created in the **/var/log** directory. When you issue this command, the specified file and all associated archive files are deleted from this directory.

Note that deleting the log file does not stop the system from logging events. If you use this command while the system is logging events, old log events will be deleted, but events after the delete operation will be recorded in the new file. To delete the file altogether, first disable logging to the file using system syslog command, and then delete it.

show log

Displays the contents of the specified log file.

Syntax

show log [**all** | **authorization** | **directory** | **file** *file-name* | **tail** [*lines*]]

Command Mode

Operational mode.

Parameters

all	Displays the contents of all master log files.
authorization	Displays all authorization attempts.
directory	Displays a listing of all user-defined log files.
file *file-name*	Displays the contents of the specified user-defined log file.
tail	Displays the last 10 lines of the system log.
lines	Specifies the number of lines that **tail** will display at the end of the system log.

Usage Guidelines

Use this command to view the contents of a log file or files.

When used with no option, this command displays the contents of the main system log, which is the default log to which the system writes syslog messages.

When used with the **authorization** option, this command displays all authorization attempts.

When used with the **directory** option, this command displays a list of all user-defined log files. Syslog messages can be written to these or to the main system log file. User-specified log files are defined using system syslog file <filename> facility <facility> level <level> command.

When **file** *file-name* is specified, this command displays the contents of the specified user-defined log file.

When used with the **tail** option, this command displays the last 10 lines of the system log file and will continue to display log messages as they are added to the log file. This command can be interrupted using <Ctrl+C>.

The *lines* option specified that the last *lines* lines of the system log are to be displayed.

show log image <image-name>

Displays the contents of the log file in the specified image.

Syntax

show log image *image-name* [**all** | **authorization** | **directory** | **file** *file-name* | **tail** [*lines*]]

Command Mode

Operational mode.

Parameters

all	Displays the contents of all master log files for the specified image.
authorization	Displays all authorization attempts for the specified image.
directory	Displays a listing of all user-defined log files for the specified image.
file *file-name*	Displays the contents of the specified user-defined log file for the specified image.
tail	Displays the last lines of the system log for the specified image.
lines	The number of lines to be displayed. If not specified, 10 lines are displayed.

Default

When used with no option, this command displays the contents of the main system log. (This is the default log to which the system writes syslog messages.)

Usage Guidelines

Use this command to view the contents of a log file or files on an image other than the current active image.

system syslog

Configures the system's syslog utility.

Syntax

set system syslog

delete system syslog

show system syslog

Command Mode

Configuration mode.

Configuration Statement

```
system {
   syslog {
   }
}
```

Parameters

None.

Default

None.

Usage Guidelines

Use this command to configure the system's syslog utility.

Using this command, you can set the destinations for log messages from different routing components (facilities) and specify what severity of message should be reported for each facility.

Log messages generated by the Vyatta system will be associated with one of the following levels of severity.

Table 7-3 Syslog message severities

Severity	Meaning
emerg	Emergency. A general system failure or other serious failure has occurred, such that the system is unusable.

Table 7-3 Syslog message severities

alert	Alert. Immediate action is required to prevent the system from becoming unusable—for example, because a network link has failed, or the database has become compromised.
crit	Critical. A critical condition exists, such as resource exhaustion—for example, the system is out of memory, CPU processing thresholds are being exceeded, or a hardware failure has occurred.
err	Error. An error condition has occurred, such as a failed system call. However, the system is still functioning.
warning	Warning. An event has occurred that has the potential to cause an error, such as invalid parameters being passed to a function. This situation should be monitored.
notice	Notice. A normal but significant event has occurred, such as an unexpected event. It is not an error, but could potentially require attention.
info	Informational. Normal events of interest are being reported as they occur.
debug	Debug level. Trace-level information is being provided.

The Vyatta system supports standard syslog facilities. These are as follows:

Table 7-4 Syslog facilities

Facility	Description
auth	Authentication and authorization
authpriv	Non-system authorization
cron	Cron daemon
daemon	System daemons
kern	Kernel
lpr	Line printer spooler
mail	Mail subsystem
mark	Timestamp
news	USENET subsystem
security	Security subsystem

Table 7-4 Syslog facilities

syslog	System logging
user	Application processes
uucp	UUCP subsystem
local0	Local facility 0
local1	Local facility 1
local2	Local facility 2
local3	Local facility 3
local4	Local facility 4
local5	Local facility 5
local6	Local facility 6
local7	Local facility 7
all	All facilities excluding “mark”

Messages are written either to the main log file (the default) or to a file that you specify. User-defined log files are written to the **/var/log/user** directory, under the user-specified file name.

The system uses standard UNIX log rotation to prevent the file system from filling up with log files. When log messages are written to a file, the system will write up to 500 KB of log messages into the file *logfile*, where *logfile* is either the main log file or a name you have assigned to a user-defined file. When *logfile* reaches its maximum size, the system closes it and compresses it into an archive file. The archive file is named *logfile***.0.gz**.

At this point, the logging utility opens a new *logfile* file and begins to write system messages to it. When the new log file is full, the first archive file is renamed *logfile***.1.gz** and the new archive file is named *logfile***.0.gz**.

The system archives log files in this way until a maximum number of log files exists. By default, the maximum number of archived files is 10 (that is, up to *logfile***.9.gz**), where *logfile***.0.gz** always represents the most recent file. After this, the oldest log archive file is deleted as it is overwritten by the next oldest file.

To change the properties of log file archiving, configure the **system syslog archive** node:

- Use the **size** parameter to specify the maximum size of each archived log file.
- Use the **files** parameter to specify the maximum number of archive files to be maintained.

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

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

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

system syslog console facility <facility> level <level>

Specifies which messages are sent to the console.

Syntax

set system syslog console facility *facility* **level** *level*

delete system syslog console facility [*facility* [**level**]]

show system syslog console facility [*facility* [**level**]]

Command Mode

Configuration mode.

Configuration Statement

```
system {
    syslog {
        console {
            facility facility {
                level level
            }
        }
    }
}
```

Parameters

facility	Multi-node. The kinds of messages that will be sent to the console. Please see the Usage Guidelines in system syslog command for supported facilities. You can send the log messages of multiple facilities to the console by creating multiple **facility** configuration nodes within the **console** node.
level	The minimum severity of log message that will be reported to the console. Supported values are **emerg**, **alert**, **crit**, **err**, **warning**, **notice**, **info**, and **debug**. Please see the Usage Guidelines in system syslog command for the meanings of these levels. By default, messages of **err** severity are logged to the console.

Default

None.

Usage Guidelines

Use this command to specify which messages are sent to the console.

Use the **set** form of this command to specify which messages are sent to the console.

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

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

system syslog file <filename> archive

Specifies the settings for log file archiving of the user-defined log file.

Syntax

set system syslog file *filename* **archive** {**files** *files* | **size** *size*}

delete system syslog file *filename* **archive** {**files** | **size**}

show system syslog file *filename* **archive** {**files** | **size**}

Command Mode

Configuration mode.

Configuration Statement

```
system {
   syslog {
      file filename{
         archive {
            files files
            size size
         }
      }
   }
}
```

Parameters

filename	Multi-node. Defines a file to which the specified log messages will be written. File names can include numbers, letters, and hyphens. Full path specifications are not accepted. You can send log messages to multiple files by creating multiple **file** configuration nodes.
files	Sets the maximum number of archive files that will be maintained for this log file. After the maximum has been reached, logs will be rotated with the oldest file overwritten. The default is 10.
size	Sets the maximum size in bytes of archive files for this log file. After the maximum has been reached, the file will be closed and archived in compressed format. The default is 1 MB.

Default

None.

Usage Guidelines

Use this command to specify the settings for log file archiving of the user-defined log file.

Use the **set** form of this command to specify the settings for log file archiving of the user-defined log file.

Use the **delete** form of this command to restore the default user-defined log file archiving configuration.

Use the **show** form of this command to view the user-defined log file archiving configuration.

system syslog file <filename> facility <facility> level <level>

Specifies which messages are sent to the user-defined log file.

Syntax

set system syslog file *filename* **facility** *facility* **level** *level*

delete system syslog file *filename* **facility** [*facility* [**level**]]

show system syslog file *filename* **facility** [*facility* [level]]

Command Mode

Configuration mode.

Configuration Statement

```
system {
   syslog {
      file filename {
         facility facility {
            level level
         }
      }
   }
}
```

Parameters

filename	Multi-node. Defines a file to which the specified log messages will be written. File names can include numbers, letters, and hyphens. Full path specifications are not accepted. You can send log messages to multiple files by creating multiple **file** configuration nodes.
facility	Multi-node. The kinds of messages that will be sent to the user-defined log file. Please see the Usage Guidelines in system syslog command for supported logging facilities. You can send the log messages of multiple facilities to this log file by creating multiple **facility** configuration nodes within the **file** configuration node.

level	The minimum severity of log message that will be reported. Supported values are **emerg**, **alert**, **crit**, **err**, **warning**, **notice**, **info**, **debug**. Please see the Usage Guidelines in system syslog command for the meanings of these levels. By default, messages of **warning** severity are logged to file.

The Vyatta system supports sending log messages to the main system log file, to the console, to a remote host, to a user-specified file, or to a user account.

Default

None.

Usage Guidelines

Use this command to specify which messages are sent to the user-defined log file.

Use the **set** form of this command to specify which messages are sent to the user-defined log file.

Use the **delete** form of this command to restore the default user-defined log file message configuration.

Use the **show** form of this command to view the user-defined log file message configuration.

system syslog global archive

Specifies the settings for log file archiving of the main system log file.

Syntax

set system syslog global archive {**files** *files* | **size** *size*}

delete system syslog global archive {**files** | **size**}

show system syslog global archive {**files** | **size**}

Command Mode

Configuration mode.

Configuration Statement

```
system {
   syslog {
      global {
         archive {
            files files
            size size
         }
      }
   }
}
```

Parameters

files	Sets the maximum number of archive files that will be maintained for the main system log file. After the maximum has been reached, logs will be rotated with the oldest file overwritten. The default is 10.
size	Sets the maximum size in bytes of archive files for the main system log file. After the maximum has been reached, the file will be closed and archived in compressed format. The default is 1 MB.

Default

None.

Usage Guidelines

Use this command to specify the settings for log file archiving of the main system log file.

Use the **set** form of this command to specify the settings for log file archiving of the main system log file.

Use the **delete** form of this command to restore the default log file archiving configuration.

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

system syslog global facility <facility> level <level>

Specifies which messages are sent to the main system log file.

Syntax

set system syslog global facility *facility* **level** *level*

delete system syslog global facility [*facility* [**level**]]

show system syslog global facility [*facility* [**level**]]

Command Mode

Configuration mode.

Configuration Statement

```
system {
   syslog {
      global {
         facility facility {
            level level
         }
      }
   }
}
```

Parameters

facility	Multi-node. The kinds of messages that will be sent to the main system log file. Please see the Usage Guidelines in system syslog command for supported facilities. You can send the log messages of multiple facilities to the main system log file by creating multiple **facility** configuration nodes within the **global** node.
level	The minimum severity of log message that will be reported. Supported values are **emerg, alert, crit, err, warning, notice, info, debug**. Please see the Usage Guidelines in system syslog command for the meanings of these levels. By default, messages of **warning** severity are logged to the main system log file.

Default

None.

Usage Guidelines

Use this command to specify which messages are sent to the main system log file.

Use the **set** form of this command to specify which messages are sent to the main system log file.

Use the **delete** form of this command to restore the default log file message configuration.

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

system syslog host <hostname> facility <facility> level <level>

Specifies which messages are sent to the remote syslog server.

Syntax

set system syslog host *hostname* **facility** *facility* **level** *level*

delete system syslog file *hostname* **facility** [*facility* [**level**]]

show system syslog file *hostname* **facility** [*facility* [**level**]]

Command Mode

Configuration mode.

Configuration Statement

```
system {
    syslog {
        host hostname {
            facility facility {
                level level
            }
        }
    }
}
```

Parameters

hostname	Multi-node. Sends the specified log messages to a host. The host must be running the syslog protocol. The *hostname* can be an IP address or a host name. Host names can include numbers, letters, and hyphens (“-”). You can send log messages to multiple hosts by creating multiple **host** configuration nodes.
facility	Multi-node. The kinds of messages that will be sent to the host. Please see the Usage Guidelines in system syslog command for supported logging facilities. You can send the log messages of multiple facilities to a host by creating multiple **facility** configuration nodes within the **host** configuration node.

level	The minimum severity of log message that will be reported. Supported values are **emerg**, **alert**, **crit**, **err**, **warning**, **notice**, **info**, **debug**. Please see the Usage Guidelines in system syslog command for the meanings of these levels. By default, messages of **err** severity are logged to hosts.

Default

None.

Usage Guidelines

Use this command to specify which messages are sent to the remote syslog server.

Use the **set** form of this command to specify which messages are sent to the remote syslog server.

Use the **delete** form of this command to restore the default remote syslog server log file message configuration.

Use the **show** form of this command to view the remote syslog server log file message configuration.

system syslog user <userid> facility <facility> level <level>

Specifies which messages are sent to the specified user’s terminal.

Syntax

set system syslog user *userid* **facility** *facility* **level** *level*

delete system syslog user *userid* **facility** [*facility* [**level**]]

show system syslog user *userid* **facility** [*facility* [**level**]]

Command Mode

Configuration mode.

Configuration Statement

```
system {
    syslog {
        user userid {
            facility facility {
                level level
            }
        }
    }
}
```

Parameters

userid	Multi-node. Sends the specified log messages to the specified user’s terminal. You can send log messages to multiple users by creating multiple **user** configuration nodes.
facility	Multi-node. The kinds of messages that will be sent to the user. Please see the Usage Guidelines in system syslog command for supported logging facilities. You can send the log messages of multiple facilities to a user account by creating multiple **facility** configuration nodes within the **user** configuration node.

level	The minimum severity of log message that will be reported to the user. Supported values are **emerg**, **alert**, **crit**, **err**, **warning**, **notice**, **info**, **debug**. Please see the Usage Guidelines in system syslog command for the meanings of these levels. By default, messages of **err** severity are logged to specified user's.

Default

None.

Usage Guidelines

Use this command to specify which messages are sent to the specified user's terminal.

Use the **set** form of this command to specify which messages are sent to the specified user's terminal.

Use the **delete** form of this command to restore the default user terminal message configuration.

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

Chapter 8: SNMP

This chapter describes the Vyatta system’s support for SNMP.

This chapter presents the following topics:

- SNMP Configuration
- SNMP Commands

SNMP Configuration

This section presents the following topics:

- SNMP Overview
- Supported MIBs
- SNMP Configuration Examples

SNMP Overview

This section presents the following topics:

- MIB Objects
- Traps
- SNMP Commands
- SNMP Versions

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 Vyatta system supports SNMP over both IPv4 and IPv6 networks.

MIB Objects

A MIB contains the set of variables/objects that are managed (for example, MTU on a network interface). Those objects are organized in a tree structure where each object is a leaf node. Each object has its unique Object Identifier (OID).

There are two types of objects: *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 will asynchronously send the traps to the SNMP managers that are configured as trap destinations or *targets*. This keeps the network manager informed of the status and health of the device.

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 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 manager to change the value of a specific object. Setting a configuration object changes the device's configuration. 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 there are three versions of SNMP:

- SNMP v1. This is the first version of the protocol. It is described in RFC 1157.
- SNMP v2. This is an evolution of the first version, and it adds a number of improvements to SNMPv1.
- SNMP v3. This version improves the security model in SNMPv2, and adds support for proxies.

The Vyatta System supports SNMP v2 with community string (SNMP v2c)

Default Object IDs

Two default object IDs set by Vyatta (within /etc/snmp/snmpd.conf) are as follows:

- sysObjectID = 1.3.6.1.4.1.30803
- sysDescr = Vyatta VSE6.1-2010.07.28

The **sysDescr** object ID can be changed using the “service snmp description <desc>” on page 326.

Supported MIBs

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

Table 8-1 lists the standard MIBs and traps supported by the Vyatta system. RFCs can be found at http://tools.ietf.org

Table 8-1 Supported Standard MIBs

MIB Name	Document Title	Notes
BGP4-MIB	RFC 1657, *Definitions of Managed Objects for the Fourth Version of Border Gateway Protocol (BGP-4)*	Protocol MIB supported plus the following traps: • bgpEstablished • bgpBackwardTransition
HOST-RESOURCES-MIB	RFC 2790, *Host Resources MIB*	
IF-MIB	RFC 2863, *The Interfaces Group MIB*	The following traps are supported: • linkUp • linkDown
IP-MIB	RFC 2011, *SNMPv2 Management Information Base for the Internet Protocol using SMIv2*	
IPV6-TC	RFC2465, *Management Information Base for IP Version 6: Textual Conventions and General Group*	
IPV6-UDP-MIB	RFC 2454, *IP Version 6 Management Information Base for the User Datagram Protocol*	
OSPF-MIB	RFC 1850, *OSPF Version 2 Management Information Base*	
RFC1213-MIB	RFC 1213, *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*	
RIPv2-MIB	RFC 1724, *RIP Version 2 MIB Extension*	
SNMPv2-MIB	RFC 3418, *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*	The following traps are supported: • coldStart • warmStart
TCP-MIB	RFC 4022, *Management Information Base for the Transmission Control Protocol (TCP)*	

Table 8-1 Supported Standard MIBs

MIB Name	Document Title	Notes
UDP-MIB	RFC 4113, *Management Information Base for the User Datagram Protocol (UDP)*	

SNMP Configuration Examples

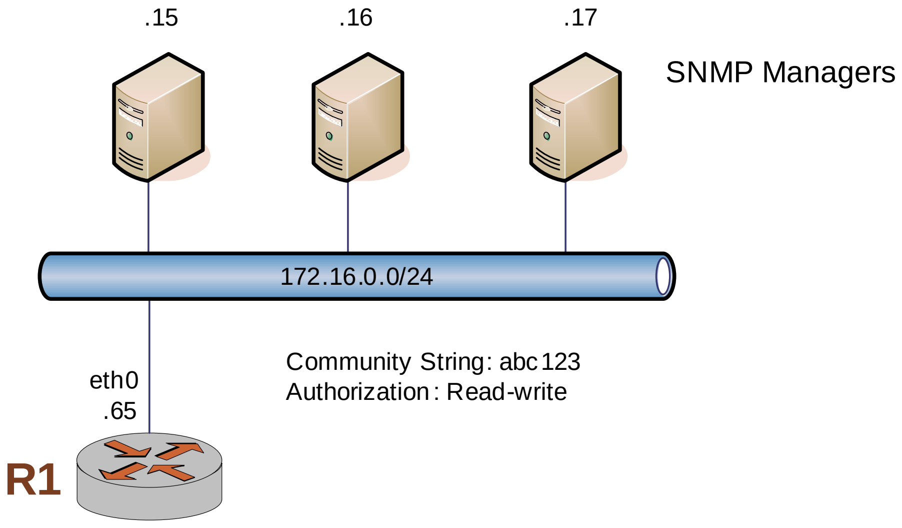
This section presents the following topics:

- Defining the SNMP Community
- Specifying Trap Destinations
- IPv6 Example

To configure SNMP, there must be at least one user created, and the Vyatta MIB model must be loaded.

This sequence sets up an SNMP community that includes three hosts, which will serve as SNMP managers, and configures the system to send traps to all three managers. When you have finished, the system will be configured as shown in Figure 8-1.

Figure 8-1 Configuring SNMP communities and traps



This section includes the following examples:

- Example 8-1 Defining an SNMP community
- Example 8-2 Specifying SNMP trap destinations

Defining the SNMP Community

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 are explicitly defined, then any client presenting the correct community string is granted read-only access to the system.
- If any client is defined, then only explicitly listed clients are granted access to the system. Those clients will have the access privilege specified by the **authorization** option. (The default is read-only.)

Example 8-1 sets the SNMP community string to abc123 and specifies three clients for the community: 176.16.0.15, 176.16.0.16, and 176.16.0.17. Read-write access is provided for this community.

To define an SNMP community, perform the following steps in configuration mode:

Example 8-1 Defining an SNMP community

Step	Command

Create the **snmp** configuration node and the **community** configuration node. Set the community string.
Navigate to the configuration node of the community.
	`vyatta@R1# set service snmp community abc123`
	`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-write.	`vyatta@R1# set authorization rw`

Commit the change, and return to the top of the configuration tree.	`vyatta@R1# commit`
	`vyatta@R1# top`

Specifying Trap Destinations

Example 8-2 directs the system to send SNMP traps to the configured network managers at 176.16.0.15, 176.16.0.16, and 176.16.0.17.

To specify trap destinations, perform the following steps in configuration mode:

Example 8-2 Specifying SNMP trap destinations

Step	Command

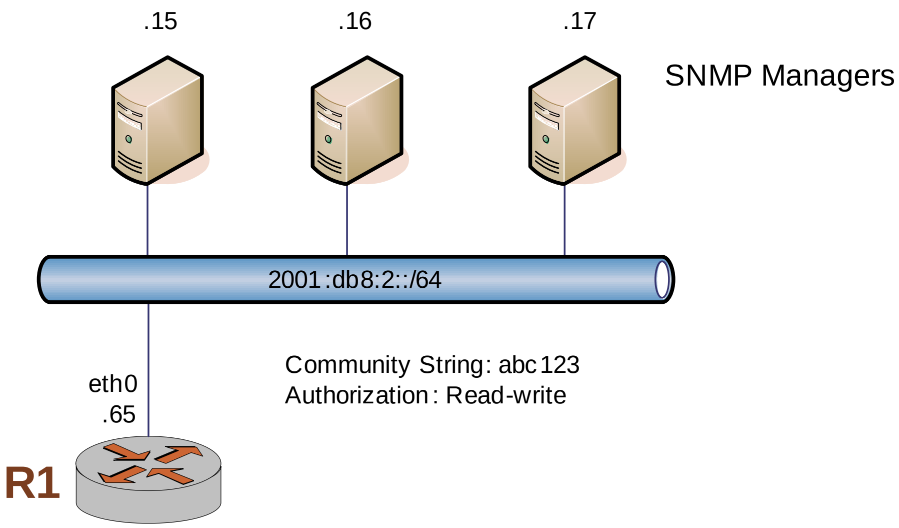
Define the trap destinations, one at a time.	`vyatta@R1# set service snmp trap-target 176.16.0.15`
	`vyatta@R1# set service snmp trap-target 176.16.0.16`
	`vyatta@R1# set service snmp trap-target 176.16.0.17`

Commit the change.	`vyatta@R1# commit`

IPv6 Example

This sequence is the same as the previous example but uses IPv6 addresses. When you have finished, the system will be configured as shown in Figure 8-2.

Figure 8-2 Configuring SNMP communities and traps - IPv6



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

Example 8-3 Defining the SNMP configurationy

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-write.	`vyatta@R1# set service snmp community abc123 authorization rw`

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`

SNMP Commands

This section presents the following commands.

Configuration Commands	
service snmp	Defines SNMP community and trap information for the Vyatta system.
service snmp community <community>	Defines an SNMP community.
service snmp community <community> authorization <auth>	Specifies the privileges this community will have.
service snmp community <community> client <addr>	Specifies the SNMP clients in this community that are authorized to access the system.
service snmp community <community> network <net>	Specifies the network of SNMP clients in this community that are authorized to access the server.
service snmp contact <contact>	Records contact information for the system.
service snmp description <desc>	Records a brief description of the system.
service snmp listen-address <addr>	Specifies the IP address the SNMP agent will listen for requests on.
service snmp location <location>	Records the location of the system.
service snmp trap-source <addr>	Specifies the IP address of the source of SNMP traps.
service snmp trap-target <addr>	Specifies the IP address of a destination for SNMP traps.
Operational Commands	
show snmp	Displays SNMP statistics.

service snmp

Defines SNMP community and trap information for the Vyatta system.

Syntax

set service snmp

delete service snmp

show service snmp

Command Mode

Configuration mode.

Configuration Statement

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

Parameters

None.

Default

None.

Usage Guidelines

Use this command to specify information about which SNMP communities this system should respond to, about the system's location and contact information, and about destinations for 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*

delete service snmp community *community*

show service snmp community *community*

Command Mode

Configuration mode.

Configuration Statement

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

Parameters

community	Optional. Multi-node. Defines 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.

Default

By default, no community string is defined.

Usage Guidelines

Use this command to specify an SNMP community.

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

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

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

service snmp community <community> authorization <auth>

Specifies the privileges this community will have.

Syntax

set service snmp community *community* **authorization** *auth*

delete service snmp community *community* **authorization**

show service snmp community *community* **authorization**

Command Mode

Configuration mode.

Configuration Statement

```
service {
   snmp {
      community community
         authorization auth
   }
}
```

Parameters

community	Optional. Multi-node. Defines 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. Specifies the privileges this community will have. Supported values are as follows: **ro**: This community can view system information, but not change it. **rw**: This community has read-write privileges. Deleting the **authorization** statement resets the privilege level to the default (**ro**).

Default

The default authorization privilege is **ro**.

Usage Guidelines

Use this command to specify the privileges this community will have.

Use the **set** form of this command to specify SNMP community privileges.

Use the **delete** form of this command to restore default SNMP community privileges.

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

service snmp community <community> client <addr>

Specifies the SNMP clients in this community that are authorized to access the system.

Syntax

set service snmp community *community* **client** *addr*

delete service snmp community *community* **client** *addr*

show service snmp community *community* **client**

Command Mode

Configuration mode.

Configuration Statement

```
service {
   snmp {
      community community
         client addr
   }
}
```

Parameters

community	Optional. Multi-node. Defines 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.
addr	Optional. Multi-node. The IPv4 or IPv6 address of an SNMP client in this community that is authorized to access the system. You can define more than one client by creating the **client** configuration node multiple times. If no client or network is defined, then any client presenting the correct community string will have read-only access to the system. If any client or network is defined then only explicitly listed clients and/or networks will have access to the system.

Default

None.

Usage Guidelines

Use this command to specify the SNMP clients in this community that are authorized to access the system.

Use the **set** form of this command to specify the SNMP clients in this community that are authorized to access the system.

Use the **delete** form of this command to remove SNMP clients in this community that are authorized to access the system.

Use the **show** form of this command to view SNMP clients in this community that are authorized to access the system.

service snmp community <community> network <net>

Specifies the network of SNMP clients in this community that are authorized to access the server.

Syntax

set service snmp community *community* **network** *net*

delete service snmp community *community* **network** *net*

show service snmp community *community* **network**

Command Mode

Configuration mode.

Configuration Statement

```
service {
   snmp {
      community community
         network net
   }
}
```

Parameters

community	Optional. Multi-node. Defines 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.
net	Optional. Multi-node. The IPv4 or IPv6 network of SNMP clients in this community that are authorized to access the server. You can define more than one network by creating the **network** configuration node multiple times. If no client or network is defined, then any client presenting the correct community string will have read-only access to the system. If any client or network is defined then only explicitly listed clients and/or networks will have access to the system.

Default

None.

Usage Guidelines

Use this command to specify an IPv4 or IPv6 network of SNMP clients in this community that are authorized to access the server.

Use the **set** form of this command to specify a network of SNMP clients in this community that are authorized to access the server.

Use the **delete** form of this command to remove a network of SNMP clients in this community that are authorized to access the server.

Use the **show** form of this command to view a network of SNMP clients in this community that are authorized to access the server.

service snmp contact <contact>

Records contact information for the system.

Syntax

set service snmp contact *contact*

delete service snmp contact

show service snmp contact

Command Mode

Configuration mode.

Configuration Statement

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

Parameters

contact	Optional. Records contact information for the system. This is stored as MIB-2 system information in the **snmpd.conf** configuration file. Letters, numbers, and hyphens are supported.

Default

None.

Usage Guidelines

Use this command to specify contact information for the system.

Use the **set** form of this command to specify 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

Command Mode

Configuration mode.

Configuration Statement

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

Parameters

desc	Optional. Records a brief description of the system. This is stored as MIB-2 system information in the **snmpd.conf** configuration file. Letters, numbers, and hyphens are supported. NOTE: When set, this text is stored as the object ID **sysDescr**. By default **sysDescr** is set to **Vyatta**.

Default

None.

Usage Guidelines

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

Use the **set** form of this command to specify a brief descriptionof 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 the SNMP agent will listen for requests on.

Syntax

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

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

show service snmp listen-address *ipv4* [**port**]

Command Mode

Configuration mode.

Configuration Statement

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

Parameters

addr	Optional. Multi-node. The IPv4 or IPv6 address the SNMP agent will listen for requests on. You can specify multiple listen addresses for SNMP by creating multiple *listen-address* configuration nodes.
port	The UDP port used for listening . The default value is 161.

Default

The SNMP agent will listen on all addresses on port 161.

Usage Guidelines

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

Use the **set** form of this command to specify the 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

Command Mode

Configuration mode.

Configuration Statement

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

Parameters

location	Optional. Records the location of the system. This is stored as MIB-2 system information in the **snmpd.conf** configuration file. Letters, numbers, and hyphens are supported.

Default

None.

Usage Guidelines

Use this command to specify the location of the system.

Use the **set** form of this command to specify 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 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 Mode

Configuration mode.

Configuration Statement

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

Parameters

addr	The IPv4 or IPv6 address of the source of SNMP traps. This address will be included source of SNMP traps in SNMP messages sent to an SNMP server. The address must an address configured on one of the system interfaces. By default the system will automatically select an IP address configured on one of the system interfaces.

Default

The SNMP trap source IP address is selected automatically.

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 of a destination for SNMP traps.

Syntax

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

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

show service snmp trap-target *ipv4* [**community** | **port**]

Command Mode

Configuration mode.

Configuration Statement

```
service {
    snmp {
        trap-target addr {
            community community
            port 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 value is **public**.
port	The destination UDP port used for trap notification. The default value is 162.

Default

None.

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 the 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.

show snmp

Displays SNMP statistics.

Syntax

show snmp

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display SNMP statistics.

Examples

Example 8-4 shows the output for **show snmp**.

Example 8-4 “show snmp”: Displaying output for 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:~$
```

Glossary

ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
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
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP

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/Ouput
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
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
ISP	Internet Service Provider
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

MIB	Management Information Base
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
ND	Neighbor Discovery
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
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation

Rx	receive
SLAAC	Stateless Address Auto-Configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SSH	Secure Shell
SSID	Service Set Identifier
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service
Tx	transmit
UDP	User Datagram Protocol
vif	virtual interface
VLAN	virtual LAN
VPN	Virtual Private Network
VRRP	Virtual Router Redundancy Protocol
WAN	wide area network
WAP	wireless access point
WPA	Wired Protected Access