DANOS-Vyatta edition

Disaggregated Network Operating System
Version 2009a

Basic System Configuration Guide

October 2020

Contents

Chapter 1. Copyright Statement

IP Infusion Inc.
3965 Freedom Circle, Suite 200
Santa Clara, CA 95054
+1 408-400-1900

http://www.ipinfusion.com/.

For support, questions, or comments via E-mail, contact:

support@ipinfusion.com.

Chapter 2. Preface

Document conventions

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

Notes, cautions, and warnings

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

hazards.

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

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

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

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

Text formatting conventions

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

Format	Description
bold text	Identifies command names. Identifies keywords and operands.
italic text	Identifies emphasis. Identifies variables. Identifies document titles.
`Courier font`	Identifies CLI output. Identifies command syntax examples.

Command syntax conventions

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

Convention	Description
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.
[ ]	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
{ **x** | **y** | **z** }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options.
x | **y**	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, *member*[*member*...].
\	Indicates a “soft” line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Chapter 3. About This Guide

This guide describes basic system concepts of DANOS Vyatta-edition, showing how to use the command-line interface, perform basic system management and monitoring tasks, manage user accounts, access system logs, and hot-plug interfaces.

Chapter 4. Using the CLI

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

Note: Configuration by using the CLI is discussed in Chapter 3: Working with Configuration .

CLI features

This section presents the following topics:

- Command modes
- Vyatta CLI and 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
- Running operational commands
- Running an operational command in configuration mode

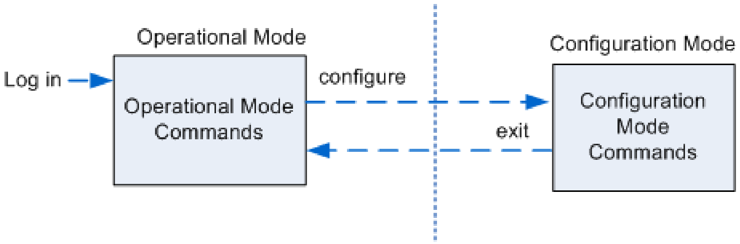
Command modes

The Vyatta CLI has two command modes: 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 in to the system, the system is in operational mode.

Figure 1: CLI command modes shows Vyatta CLI command modes.

Figure 1. CLI command modes



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

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

Entering the `exit` command in operational mode logs you off the system.

Vyatta CLI and system shell

The CLI of the router includes two kinds of commands:

- Commands for operating and configuring the router
- 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 for which you have the privileges to execute, including operating system commands, can be executed from within the Vyatta CLI.

Accessing the CLI

To access the CLI, you log in to the router, either directly through the VGA console, a serial console, or remotely by using a Secure Shell (SSH) or Telnet session. The VGA console also provides nine virtual console sessions. These virtual consoles (tty1 through tty9) can be accessed by using the key combinations ALT-F1 (for tty1) through ALT-F9 (for tty9). tty1 through tty6 provide a login prompt. tty7 through tty9 are not used.

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

```
vyatta login:
```

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

Note: You can change user accounts by using operating system commands, but the changes do 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 vyatta user. The default password for the vyatta account is vyatta. The vyatta user has administrator-level privileges and can execute all router commands and all operating system commands. Note that, although the user can execute both router commands and operating system commands, command completion and CLI help show only router commands for clarity.

User privilege levels

The router supports two user roles:

- Admin users
- Operator users

Admin users

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

The vyatta default user is an admin user.

To create an admin user, enter 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 employ 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 makes it very difficult to distinguish available 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 setting removes the restriction on command completion for all users, regardless of privilege level.

Operator users

Operator users have read-only access to configuration plus the ability to execute router operational commands. Operator users can view in operational mode (by using `show` commands), configure their terminal settings (by using the `set terminal` command), and exit from the Vyatta CLI (by using the `exit` command). Operator users cannot enter configuration mode; however, they can display configuration by entering 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, enter 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 by using command completion for operator-level users is restricted to router 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 setting removes the restriction on command completion for all users, regardless of privilege level.

Command prompts

The command prompt shows you the user account under which you are logged in, the host name of the system you are logged in to, and whether you are in configuration mode or operational mode.

The format of the command prompt in configuration mode is 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 in and *hostname* is the host name configured for the system; see Table 1: Command prompts for examples.

Table 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 dataplane
```

In this example, the space characters separate the command line into three components: `show`, `interfaces`, and `dataplane`.

If you want to enter a string of characters 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 character string that includes a space, you must enclose the string in double quotation marks, as shown below.

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

In this example, the space within the character string external inbound is within quotation marks 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 command to the left of the pipe should be processed by using the command to the right of the pipe, as shown in the following example.

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

In this example, the pipe character tells the shell to run the `show interfaces` command and then process the output by using the `match dp` command; as a result, only lines that contain the dp character string are 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 (&)
- semicolon (;)
- comma (,)
- left parenthesis (()
- right parenthesis ())
- left angle bracket (<)
- right angle bracket (>)
- backslash (\)
- pound sign (#)

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

Note that within a quotation-enclosed string, you can include a literal quotation mark by preceding it with a backslash, as shown in the following example.

```
"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 quotation marks or backslashes as literal configuration values.

Command completion

To save keystrokes, the system accepts unambiguous command prefixes in place of the full command. For example, typing `sh configu` in operational mode is equivalent to typing `show configuration`.

You can also have the system automatically complete a command syntax by entering or pressing any of the following at the command prompt.

Table 2. CLI help keystrokes

Enter or press this:	To display this:
<Tab>	Automatic completion of 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 possible tokens. Pressing <Tab> a second time displays command help for each possible token. (Note that the space following a command or keyword counts as a token.)

Table 2. CLI help keystrokes (continued)

Enter or press this:	To display this:
? or <Alt>-?	The set of possible tokens. Pressing ? a second time displays command help for each possible token. **Note:** To enter a literal question mark, first enter <Ctrl>+v, then the question mark.

The following example shows how to find all available commands.

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

The following example shows how to request command completion for the `sh` entered character string. In this example, the command to be completed is unambiguous.

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

vyatta@R1~$ show
```

The following example shows how to request command completion for the `s` entered character string. In this case, more than one command can 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 enclosed in double quotation marks. When used within double quotation marks, the <Tab> key generates a tab character and the <Alt>+? key combination generates a question mark (?) character.

In configuration mode, the following symbols are displayed next to nodes in their completion help text to indicate the node type.

Symbol	Node
+	Multinode
>	Nonleaf node
+>	Tag node (multiple nonleaf)

The following example shows the node symbols next to possible completions for the `interfaces dataplane` command.

```

vyatta@vyatta:~$ configure

[edit]
```

```
vyatta@vyatta# set interfaces dataplane dp0p0p1
<Tab>
Possible Completions:
   <Enter>              Execute the current command
+  address              IP address
 > bridge-group         Add this interface to a bridge group
   description          Description
 > dhcpv6-options       DHCPv6 options
   disable              Disable interface
   disable-link-detect Ignore link state changes
 > firewall             Firewall options
 > flow-monitoring      Flow-Monitoring configuration for interface
 > ip                   <No help text available>
 > ipv6                 IPv6 parameters
   log_martians         Enable the logging of bogus packets
   mac                  Media Access Control (MAC) address
   mtu                  Maximum Transmission Unit (MTU)
 > policy               PBR Options
   qos-policy           Qos policy for interface
   sflow                Enable/Disable sflow for interface
+> vif                  Virtual Interface (VIF) ID
 > vrrp                 Virtual Router Redundancy Protocol (VRRP)
 > xconnect             Specify the parameters for cross-connect
```

Command history

The router shell supports a command history in which the commands that you run are stored in an internal buffer and can be run or edited.

Table 3: Command history keystrokes shows the most important history keystrokes.

Table 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 router shell supports Emacs-style command editing.

Table 4: Command-line editing keystrokes shows the most important editing keystrokes.

Table 4. Command-line editing keystrokes

Type this	To do this
<Left Arrow> <Control>-b	Move backward in the command line.
<Right Arrow> <Control>-f	Move forward in the command line.
<Control>-a	Move to the beginning of the command line.
<Control>-e	Move to 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 shows the “more” indication where the information breaks.

Table 5: Display options within a "more" screen shows the keystrokes for controlling the display of information in a “more” screen.

Table 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.
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.

Table 5. Display options within a "more" screen (continued)

Type this	To do this
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 router 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 by using the pipe, or vertical bar, operator (|).

Table 6: "pipe" filter commands shows the pipe commands implemented for the router.

Table 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`	Do not 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 run 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 by 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#
```

Chapter 5. Basic commands for using the CLI

copy file to

Copies a file or directory.

```
copy file  from-file to  to-file
```

from-file

The source file or directory.

to-file

The destination file or directory.

Operational mode.

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 refers 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.

A file or directory can be copied on the local machine. Only a file can be copied to and from the remote machine by using FTP, SCP, or TFTP.

Note: Use this command with caution because its effects are not reversible. In addition, when downloading files, IP Infusion Inc. recommends that you use SCP, TFTP, FTP, or HTTP.

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

Table 7. Specifying file locations

Location	Specification
FTP server	`ftp://user:passwd@host/file` where user is the username on the host, passwd is the password associated with the username, 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 username on the host, passwd is the password associated with the username, 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.

The following example shows how to copy the contents of the `/config/x509/` directory on the currently running system to the `/config/x509/` directory of the TEST-IMAGE-1 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 0.96
vyatta@vyatta:~$
```

delete file

Deletes a file or directory.

`delete file` *`file`*

file

A file or directory to delete, including the path.

Operational mode.

Use this command to delete a file or directory.

This command is optimized for configuration files and directories in that command completion refers 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 because its effects are not reversible.

This example shows how to delete the `/config/user-data/xxx` file from the currently running system.

```
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.

`exit`

Operational mode.

Use this command in operational mode to exit the system.

run

Runs an operational command without leaving configuration mode.

`run` *`command`*

command

An operational command to be run.

Configuration mode.

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

This example shows how to run the show date command (an operational command) from configuration mode.

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

show file

Displays information about a file or directory.

`show file` *`file`*

file

A file or directory about which to display information.

Operational mode.

Use this command to display information about a file or directory.

This command is optimized for configuration files and directories in that command completion refers 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 is displayed for various file types, as shown in the following table.

Table 8. Types of information displayed for various file types

File Type	Information Displayed
Directory	Directory contents
Text file	Information about the file and file contents
Packet capture file (*.pcap)	Information about the file and file contents in the form of a packet capture from tshark
Binary file	Information about the file and file contents in the form of a hexadecimal dump

The following example shows how to display the contents of the `/config` directory on the currently running system.

```
vyatta@vyatta:~$ show file running://config
########### DIRECTORY LISTING ###########
total 36K
drwxrwsr-x 1 root 4.0K Mar 21 17:21 archive/
drwxrwsr-x 1 root 4.0K Mar 21 07:56 auth/
drwxrwsr-x 1 root 4.0K Mar 21 07:56 scripts/
drwxrwsr-x 1 root 4.0K Mar 21 07:56 support/
drwxr-sr-x 1 root 4.0K Mar 21 07:57 url-filtering/
drwxrwsr-x 1 root 4.0K Mar 21 07:56 user-data/
-rwxrwxr-x 1 root 1.9K Mar 21 17:21 config.boot
-rwxrwxr-x 1 root 4.2K Mar 20 17:14 webgui2_default_config.boot
vyatta@vyatta:~$
```

This example shows how to display partial contents of the `/tmp/test1.pcap` file on the currently running system.

```
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 (dataplane, 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
...
```

This example shows how to display partial contents of the `/config/r1.tar` file on the currently running system.

```

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 6. Working with Configuration

This chapter describes utilities for configuration management on the router.

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

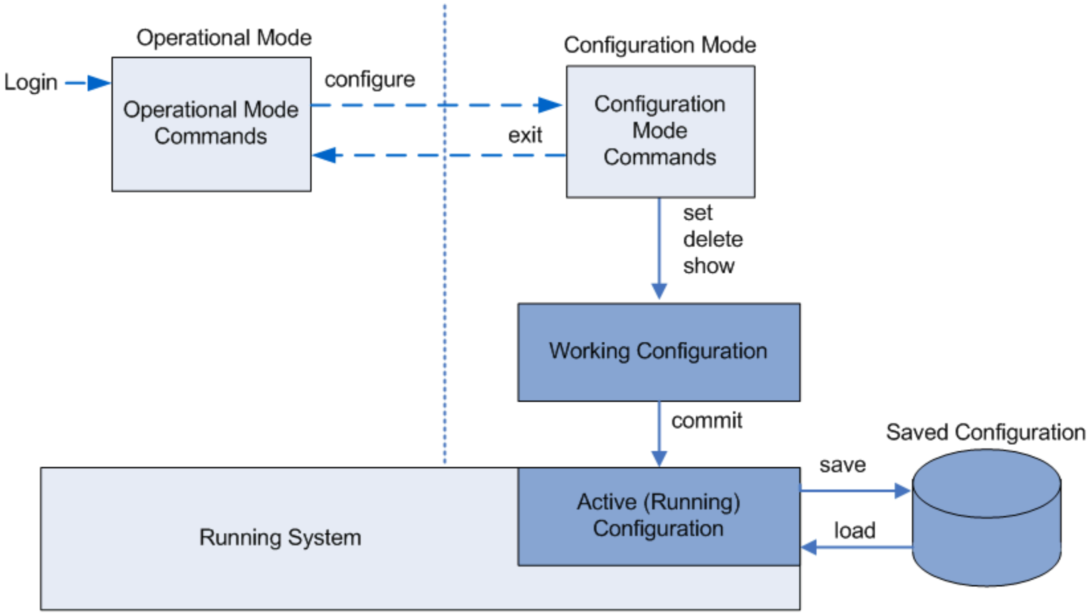
Terminology

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

- Active or “running” configuration. This configuration is the one that is 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 (by using the `save` command), it is saved to the `config.boot` file in the `/config` directory of the local system. When you reboot, the system reads and loads the configuration from `config.boot`.

Figure 2: CLI configuration states shows configuration states possible in the Vyatta CLI.

Figure 2. CLI configuration states



Location of configuration information

Boot configuration is stored in the `config.boot` file 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, make certain that any certificate file you reference within a configuration node is stored here.
- `scripts`. This directory stores scripts referenced from within the configuration nodes; for example, VRRP transition scripts. To ensure smooth upgrades, and to preserve this kind of information across upgrades, make certain 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, make certain 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 make changes to them only with great care.

Configuration hierarchy

router 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. Three kinds of statements exist:

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

From a system perspective, 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 nonempty, 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 by using the `commit` command or discard them by 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 looks like this:

```
[edit]
```

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

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

Table 9: Commands for navigating in configuration mode shows the commands for navigating in configuration mode.

Table 9. 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 in which you are interested and run commands relative to your location. This navigation saves typing if you need to work on a particular part of the configuration hierarchy.

The following example shows how to navigate to the configuration node for the dp0p1p3 data plane interface. After you have navigated to the node, you can show configuration directly without specifying the full path.

```
vyatta@R1# edit interfaces dataplane dp0p1p2
[edit interfaces dataplane dp0p1p2]
vyatta@R1# show
 hw-id 00:13:46:e6:f6:87
[edit interfaces dataplane dp0p1p3]
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 how to display configuration for all configured interfaces.

```
vyatta@R1# show interfaces
    dataplane dp0p1p1 {
       address 10.1.0.62/24
       hw-id 00:40:63:e2:e4:00
    }
    dataplane dp0p1p2 {
        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 how to display configuration for only the dp0p1p1 data plane interface.

```
vyatta@R1# show interfaces dataplane dp0p1p1
 address 10.1.0.62/24
 hw-id 00:40:63:e2:e4:00
```

When the display is too large for one screen, the display stops after one screen is shown. In this case, press one of the following keys to perform the indicated action.

- <Enter> to display the next line
- <Space> to display the next screen
- <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 by using the `show configuration` command, as in the following example.

```
vyatta@R1:~$ show configuration
interfaces {
    dataplane dp0p1p1 {
        address 192.168.1.77/24
        hw-id 00:0c:29:68:b3:9f
    }
    dataplane dp0p1p2 {
        hw-id 00:0c:29:68:b3:a9
    }
    loopback lo {
    }
}
service {
    ssh {
     }
}
```

Changing configuration information

This section presents the following topics:

- Adding or modifying configuration
- Deleting configuration
- Committing configuration changes
- Discarding configuration changes

Adding or modifying configuration

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

```
vyatta@R1# set interfaces dataplane dp0p1p3 address 192.168.1.100/24
vyatta@R1#
```

Then use the `show` command to see the change.

```
vyatta@R1# show interfaces dataplane dp0p1p3
+address 192.168.1.100/24
 hw-id 00:13:46:e6:f6:87
vyatta@R1#
```

Notice the plus sign (+) 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 by using the `commit` command.

Another option is to use the `compare` command to see the change.

```
vyatta@R1# compare
[edit interfaces dataplane dp0p1p3]
+address 192.168.1.100/24
vyatta@R1#
```

You can change 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 modify or add configuration.

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 dataplane dp0p1p2 address 192.168.1.100/24
```

Then use the `show` command to see the change.

```
vyatta@R1# show interfaces dataplane dp0p1p3
-address 192.168.1.100/24
 hw-id 00:13:46:e6:f6:87
vyatta@R1#
```

Notice the minus sign (-) 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 by using the `commit` command.

Another option is to use the `compare` command to see the change.

```
vyatta@R1# compare
[edit interfaces dataplane dp0p1p3]
-address 192.168.1.100/24
vyatta@R1#
```

Some configuration nodes are mandatory; these nodes 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 a router, configuration changes do not take effect until you commit them by using the `commit` command.

```
vyatta@R1# commit
```

A line that contains uncommitted changes is flagged as follows:

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

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

```
vyatta@R1# show interfaces dataplane dp0p1p3
-address 192.168.1.100/24
 hw-id 00:13:46:e6:f6:87
vyatta@R1# commit
vyatta@R1# show interfaces dataplane dp0p1p3
 hw-id 00:13:46:e6:f6:87
vyatta@R1#
```

Note: When you commit changes in the configuration mode, the changes are saved to the startup configuration. As a result, the changes are preserved even after a reboot.

CAUTION: If your login username is not a member of the "secrets" login user group and you either save a configuration through the REST API or use the `save` command, the encrypted passwords in the configuration file are replaced with the ******** placeholder. If you load this configuration, the replaced password fields trigger validation errors because the placeholder does not match the format for an encrypted password. Do not commit this configuration. If you ignore the error message and perform a commit with this invalid configuration, the passwords are deleted.

Discarding configuration changes

You cannot exit from configuration mode with uncommitted configuration changes; you must either commit the changes or discard them. If you do not want to commit the changes, you can discard them by 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:~$
```

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
- Cloning configuration across system images
- Performing file operations on configuration files and directories

Saving the running configuration

You can save the configuration to file for your own use; for example copying to another device, by using the `save <filename>` command in configuration mode.

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

The running configuration is saved automatically when you enter commit or commit-confirm. Entering the `save` command without a filename has no effect.

```
vyatta@R1# save
Saving configuration to '/config/config.boot'...
'commit' saves configuration. This command has no effect
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 the file to a hard drive, flash memory, 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.

⚠ **CAUTION:** If your login user is not a member of the login user group "secrets" and you save a configuration either through the REST API or use the `save` command, the encrypted passwords in the configuration file are replaced with the ******** placeholder. If you load this configuration, the replaced password fields trigger validation errors because the placeholder does not match the format for an encrypted password. Do not commit this configuration. If you ignore the error message and perform a commit with this invalid configuration, the passwords are deleted.

Table 10: Specifying locations for the configuration file shows how to specify the syntax for files from different file locations when you save files in configuration mode.

Table 10. 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.
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 are prompted for a user name and password.
HTTP server	Use the following syntax for *file-name*: http://*ip-address* /*config-file* where *ip-address* is the IP address of the HTTP server, and *config-file* is the configuration file, including the path.

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 `/config` configuration directory.

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

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

A loaded configuration then needs to be committed to become the active configuration.

⚠ **CAUTION:** If your login user is not a member of the login user group "secrets" and you save a configuration either through the REST API or use the `save` command, the encrypted passwords in the configuration file are replaced with the ******** placeholder. If you load this configuration, the replaced password fields trigger validation errors because the placeholder does not match the format for an encrypted password. Do not commit this configuration.
If you ignore the error message and perform a commit with this invalid configuration, the passwords are deleted.

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 the `config.boot` file in the default `/config` directory.

Merging saved and running configurations

You can merge a saved configuration with the active (running) configuration by using the `merge` command. An example is provided in Example 3-16 .

The merger 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.

Rolling back to a previous version

You can roll back system configuration to any archived version by using the `rollback` command.

To see a list of available configuration file revisions, use the `show system commit` command in operational mode.

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 the `config.boot` file in the `/config` directory. 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 form of *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 by 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 by using the `system config-management commit-archive` command. FTP, SCP, and TFTP destinations are supported.

Comparing configuration versions

You can compare two versions of configuration by using the `show system commit` and `compare` commands. Table 11: Commands for comparing configuration versions summarizes options for comparing configuration versions.

Table 11. Commands for comparing configuration versions

Use this command	To see the
Configuration Commands	

Table 11. Commands for comparing configuration versions (continued)

Use this command	To see the
`compare`	Difference between the working and active configuration.
`compare` *n*	Difference between the working configuration and revision *n*.
`compare` *n m*	Difference between revision *n* and revision *m*.
Operational Commands	
`show system commit`	Summary of commits.
`show system commit file` *n*	Full configuration at revision *n*.
`show system commit file` *n* `compare` *m*	Difference between revision *n* and revision *m*.
`show system commit diff` *n*	What changed in a given commit (between revision *n* and revision *n* +1). This command is equivalent to the `show system file`*n*`compare`*n* +1 command.

Cloning configuration across system images

You can copy the `/config` directory from one image to another by 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 because 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 router 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 2: Using the CLI .

These commands are optimized for configuration files and directories because command completion refers 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.

Chapter 7. Configuration Commands

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

Clones the configuration directory of one image to another image.

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

The configuration directory is copied from the running system.

dest-image-name

The name of the image to which the configuration directory is copied.

source-image-name

Optional. The name of the image from which the configuration directory is copied.

Operational mode

Use this command to copy the configuration (`/config`) directory from one image to another. By default, the source image is the currently running image.

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

Note: Use this command with caution because it 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 the directory to the running image or the disk-installation image.

The following example shows how to copy the contents of the `/config` directory of the currently running system to the `/config` directory of the TEST-IMAGE-1 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 severity level of replaced with the data from running://
Do you wish to continue? (Y/N): y
```

```
config/
config/.vyatta_config
...
```

The following example shows how to copy the contents of the `/config` directory of the TEST-IMAGE-2 system to the `/config` directory of the TEST-IMAGE-1 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 severity level of replaced with the data from TEST-IMAGE-2
Do you wish to continue? (Y/N): y
sending incremental file list
config/
config/.vyatta_config
...
```

commit

The changes are copied to the startup configuration automatically. As a result, the changes are preserved even after a reboot.

`commit` **`comment`** *`comment-text`*

comment-text

Text that describes the reason for the commit.

Configuration mode

Use this command to apply and save uncommitted changes to the configuration.

When you add configuration to, modify existing configuration in, or delete configuration from the system, the changes you make must be committed before they take effect. To commit changes, use the `commit` command.

If you try to exit or quit configuration mode while uncommitted configuration changes still exist, the system gives you a warning. You cannot exit configuration mode until you either commit the changes by entering the `commit` command or discard the changes by using the `discard` command.

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

⚠ **CAUTION:** If your login username is not a member of the "secrets" login user group and you either save a configuration through the REST API or use the `save` command, the encrypted passwords in the configuration file are replaced with the ******** placeholder. If you load this configuration, the replaced password fields trigger validation errors because the placeholder does not match the format for an encrypted password. Do not commit this configuration. If you ignore the error message and perform a commit with this invalid configuration, the passwords are deleted.

Committing changes can take time, depending on the complexity and activity of the system. Be prepared to wait for several minutes for the system to complete committing the changes.

If two or more users are logged into the system in configuration mode and one user changes the configuration, the other user or users receive a warning.

Note: Commits are logged at the logging levels of **info** and **debug**.

The following example displays a commit command that includes the system commit history and shows that the changes are automatically saved.

```
vyatta@vyatta# show system commit
0   2017-01-06 09:48:45 by vyatta
    Set loopback interface address
1   2017-01-06 09:48:20 by vyatta
    Update loopback interface description

vyatta@vyatta# set interfaces loopback lo address 2.2.2.2/32
[edit]
vyatta@vyatta# commit comment "Change loopback interface address"
[edit]
vyatta@vyatta# run show system commit
0   2017-01-06 09:50:15 by vyatta
    Change loopback interface address
1   2017-01-06 09:48:45 by vyatta
    Set loopback interface address
2   2017-01-06 09:48:20 by vyatta
    Update loopback interface description
```

commit-confirm

Commits and saves the configuration as specified with the `commit` command. The commit-confirm command rolls back the configuration to the last saved configuration if a confirmation is not provided within the given timeout period.

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

minutes

The time, in minutes, to wait for the confirmation to be provided.

comment *comment*

Specifies a comment to appear in the revision history for the configuration file. The format is a character string enclosed in double quotation marks.

Configuration mode

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 reconnect to the system, for example, accidentally changing the IP address of the management port.

A reboot during the commit-confirm timeout window results in the restoration of the previous configuration.

After making a configuration change, enter the `commit-confirm` command, specifying the confirmation interval. Commit the change. If the commit is completed, without incident, confirm the commit by entering the command confirm. If you do not confirm, the changes are rolled back.

If the new commit-confirmed configuration causes a crash before the confirmation window expires, the system reboots and rolls back to avoid an endless cycle of rebooting.

If you attempt to use the GRUB configuration recovery option when rebooting during the commit-confirm timeout window is active, a message to wait for the rollback to be completed is displayed.

If you enter `commit`, `commit-confirm`, or `rollback` commands during the commit-confirm timeout, the current commit that is pending confirmation is implicitly confirmed. For example, `rollback 1` takes you back to the immediately preceding commit (before the original commit-confirm), and `rollback 0` recommits the current configuration.

The following example shows how to save commit-confirm changes.

```

vyatta@R1# set interfaces loopback lo description "updated description"
[edit]
vyatta@R1# commit-confirm 2 comment "Updated loopback interface
 description"

commit will rollback to previous version in 2 minutes unless you enter
 'confirm'

```

If you enter `confirm` within the timeout period of 2 minutes, the system configuration corresponds to the following configuration:

```
vyatta@R1# run show system commit
0   2017-01-06 10:20:14 by vyatta
    Updated loopback interface description
1   2017-01-06 09:50:15 by vyatta
    Change loopback interface address
2   2017-01-06 09:48:45 by vyatta
    Set loopback interface address
3   2017-01-06 09:48:20 by vyatta
    Update loopback interface description
...
```

If you do not enter `confirm` within the timeout period of 2 minutes, the system configuration rolls back to the previous version as illustrated here:

```
vyatta@R1# run show system commit
0   2017-01-06 10:22:15 by configd
    Rollback to previous version.
1   2017-01-06 10:20:14 by vyatta
    Updated loopback interface description
2   2017-01-06 09:50:15 by vyatta
    Change loopback interface address
3   2017-01-06 09:48:45 by vyatta
    Set loopback interface address
4   2017-01-06 09:48:20 by vyatta
    Update loopback interface description
```

compare

Compares two sets of configuration information.

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

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.

rev-num

A configuration file revision to be compared.

rev-num1

Another configuration file revision to be compared.

Configuration mode

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

You can see the list of configuration file revisions by using show system commit in operational mode (use `run show system commit` in configuration mode).

The following example shows the working and active configurations being compared on R1.

```
vyatta@R1# compare
[edit system]
+options {
+       reboot-on-panic true
+}
[edit]
vyatta@R1#
```

configure

Enters configuration mode.

`configure`

Operational mode

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 from ~$ to # to mark the change in command mode.

The following example shows the system response to entering configuration mode. In this example, notice that the command prompt changes from ~$ to # when configuration mode is entered.

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

confirm

Confirms the previous commit-confirm change and cancels the automatic rollback.

`confirm`

Configuration mode

Use this command to confirm a successful change in configuration 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 command commit-confirm. This command sets the system to wait for confirmation that a configuration has succeeded.

Entering the `confirm` command within the specified commit-confirm interval causes the configuration change to be accepted. If confirmation is not provided by entering this command, the system reboots to the previous configuration.

The following example shows how to confirm `commit-confirm` changes.

```
vyatta@R1# set int loopback lo description anotherTest
[edit]
vyatta@R1# commit-confirm 2 comment anotherTest
```

```
commit will rollback to previous version in 2 minutes unless you enter
 'confirm'


[edit]
vyatta@R1# confirm
[edit]
vyatta@R1#
```

If you enter `confirm` within the timeout period of 2 minutes, the system configuration corresponds to the following version:

```
vyatta@R1# run show system commit
0   2017-01-06 10:20:14 by vyatta
    Updated loopback interface description
1   2017-01-06 09:50:15 by vyatta
    Change loopback interface address
2   2017-01-06 09:48:45 by vyatta
    Set loopback interface address
3   2017-01-06 09:48:20 by vyatta
    Update loopback interface description
...
```

If you do not enter `confirm` within the timeout period of 2 minutes, the system configuration rolls back to the previous version, as illustrated here:

```

vyatta@R1# run show system commit
0   2017-01-06 10:22:15 by configd
    Rollback to previous version.
1   2017-01-06 10:20:14 by vyatta
    Updated loopback interface description
2   2017-01-06 09:50:15 by vyatta
    Change loopback interface address/
3   2017-01-06 09:48:45 by vyatta
    Set loopback interface address
4   2017-01-06 09:48:20 by vyatta
    Update loopback interface description
```

delete

Deletes a configuration node.

`delete` *`config-node`*

config-node

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

Configuration mode

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 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.

The following example shows how to delete a DNS server from system 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 changes to configuration.

`discard`

Configuration mode

Use this command to discard all uncommitted changes to configuration.

The following example shows an uncommitted deletion and an uncommitted addition that are then discarded. In the example, notice that the uncommitted deletion is flagged with a minus sign “-” and the uncommitted addition is flagged with a plus sign (+), which disappear after the `discard` command is entered.

```
vyatta@vyatta# show interfaces dataplane dp0p1p3
-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 dataplane dp0p1p3
 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.

`edit` *`path`*

path

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

Configuration mode

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

In the following example, the user 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 the example, notice that the prompt changes to [edit system login] to mark the location 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 usage.

`exit` [**discard**]

discard

Exits configuration mode to operational mode and discards all uncommitted changes.

Configuration mode.

Operational mode

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 gives you a warning. You cannot exit from configuration mode until you either commit the changes by entering the `commit` command or discard the changes by using the **discard** option. This option applies only to this usage.

Use this command in operational mode to exit the system.

load

Loads from a file a configuration that was previously saved.

```
load  file-name
```

file-name

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

Configuration mode

Use this command to load from a file a configuration that was previously saved.

The loaded configuration becomes the working configuration and must be committed before it becomes the active configuration.

⚠ **CAUTION:** If your login user is not a member of the login user group "secrets" and you save a configuration either through the REST API or use the `save` command, the encrypted passwords in the configuration file are replaced with the ******** placeholder. If you load this configuration, the replaced password fields trigger validation errors because the placeholder does not match the format for an encrypted password. Do not commit this configuration.
If you ignore the error message and perform a commit with this invalid configuration, the passwords are deleted.

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. In addition, an error is reported if an invalid configuration file is loaded.

The default configuration directory is `/config`.

Note: When downloading files, IP Infusion Inc. recommends that you use SCP, TFTP, FTP, or HTTP.

The following table shows how to specify the syntax for files from different file locations.

Table 12. 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 username on the host, *passwd* is the password associated with the username, *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 are prompted for them.
SCP server	Use the following syntax for *file-name*: scp://*user:passwd@host /config-file* where *user* is the username on the host, *passwd* is the password associated with the username, *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 are 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.

The following example shows how to load the `testconfig` file from the default configuration directory.

```
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.

`merge` ***`file-name`***

file-name

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

Configuration mode

Use this command to load from a file a configuration that was previously saved and merge it with the active (running) configuration. The merger adds new configuration entries and applies any modifications to existing active entries to produce a new working configuration. This 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.

The default configuration directory is `/config`.

The following table shows how to specify the syntax for files from different file locations.

Table 13. 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 username on the host, *passwd* is the password associated with the username, *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 are prompted for them.
SCP server	Use the following syntax for *file-name*: scp://*user:passwd@host* /*config-file* where *user* is the username on the host, *passwd* is the password associated with the username, *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 are 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.

The following example shows how to load the testconfig configuration file from the default configuration directory and merge it with the active configuration.

The new working configuration must be committed before it becomes active. After the merger, 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`.

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

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

monitor command

Monitors the command output of a show command.

`monitor command` *`show-command`*

`run monitor command` *`show-command`*

show-command

Any `show` command to be monitored. The `show` command must be enclosed in quotation marks.

Operational mode.

Configuration mode

Use this command to display the output of a `show` command. The session stays open and display information is refreshed every two seconds.

Use the `run` version of this command in configuration mode.

rollback

Allows you to roll back configuration to a specific revision.

`rollback` *`rev-num`* **`comment`** *`comment-text`*

rev-num

The configuration revision to roll back to.

comment-text

Comment text describes the reason for rollback.

Configuration mode

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).

This example allows you to roll back existing configuration to the specified revision of the router configuration.

```
vyatta@vyatta# rollback
Possible completions:
  <N>   Rollback to revision N
  0     2016-09-13 17:32:07 vyatta
  1     2016-09-13 17:19:06 vyatta
  2     2016-09-13 17:09:37 vyatta
  3     2016-09-13 17:07:04 configd
  4     2016-09-13 16:43:11 configd

[edit]
vyatta@vyatta# rollback 0
Proceed with reboot? [confirm][y]

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

save

Saves the running configuration to a file.

`save` *`file-name`*

file-name

The name of a file in which the information is to be saved, including the path to the file.

Configuration mode

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 by using load. A nonabsolute path is interpreted relative to the default configuration directory, which is `/config`.

The following table shows how to specify the syntax for files from different file locations.

Table 14. 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*

Table 14. Specifying locations for the configuration file (continued)

Location	Specification
	where *user* is the username on the host, *passwd* is the password associated with the username, *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 are prompted for them.
SCP server	Use the following syntax for *file-name*: scp://*user:passwd@host /config-file* where *user* is the username on the host, *passwd* is the password associated with the username, *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 are 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 write over `my-config.boot`, the system moves the previous file to `my-config.boot~`.

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

CAUTION: If your login user is not a member of the login user group "secrets" and you save a configuration either through the REST API or use the `save` command, the encrypted passwords in the configuration file are replaced with the ******** placeholder. If you load this configuration, the replaced password fields trigger validation errors because the placeholder does not match the format for an encrypted password. Do not commit this configuration. If you ignore the error message and perform a commit with this invalid configuration, the passwords are deleted.

The following example shows how to save the running configuration to the `my-config` file in the default configuration directory, exit configuration mode, and display the set of files stored in the configuration directory.

```
vyatta@vyatta# save my-config
Saving configuration to '/config/my-config'...
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:~$
```

The following example shows how to save the current running configuration to the `my-config` file in the root directory of a TFTP server at 10.1.0.35.

```
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 an attribute in an existing configuration node.

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*]

config-node

A 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 a configuration node. The identifier is mandatory if the configuration node has an identifier; otherwise, it is not allowed.

attribute

A configuration attribute to be set. If the attribute statement does not exist, it is created. If the attribute statement already exists, its value is set to the new value.

value

The new value of the attribute. The value is mandatory if the attribute statement requires a value; otherwise, it is not allowed.

Configuration mode

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 by using commit.

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

The following example shows how to add a configuration node for a data plane interface and commit the change.

```
vyatta@vyatta# set interfaces dataplane dp0p1p2 address
 192.150.187.108/24
vyatta@vyatta# commit
```

show

Displays configuration information in configuration mode.

```
show  [ -all ] config-node
```

config-node

A configuration node you want to display, including the path. The node must exist and the created node must have been committed.

Specification of the configuration node is interpreted relative to your current position in the configuration tree.

-all

Includes default information in the displayed information.

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

When used without the **-all** keyword, this command does not display default information.

Configuration mode

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

This command displays the specified configuration node and all subnodes. 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, a number of `show` commands are available in operational mode.

The following example shows how to display the configuration information of data plane interfaces by using the `show` command in configuration mode. In this case, because the **-all** keyword is not used, the default information is not included in the output.

```
vyatta@vyatta# show interfaces dataplane
 dataplane dp0s160 {
        address 10.18.170.205/24
 }
[edit]
```

The following example shows how to display the configuration information, including the default information, of data plane interfaces by using the `show` command with the **-all** keyword in configuration mode.

```
vyatta@vyatta# show -all interfaces dataplane
 dataplane dp0s160 {
        address 10.18.170.205/24
        ip {
                gratuitous-arp-count 1
                rpf-check disable
        }
        ipv6 {
                dup-addr-detect-transmits 1
        }
        mtu 1500
        vlan-protocol 0x8100
 }
[edit]
```

The following example shows how to display the configuration information, including the default information, of the SSH service by using the `show` command with the **-all** keyword in configuration mode.

```
vyatta@vyatta# show -all service ssh
 ssh {
        authentication-retries 3
        timeout 120
 }
[edit]
```

show configuration (operational)

Displays system configuration from operational mode.

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

Displays only the values that have been set explicitly, that is, nondefault values.

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 commands generate the configuration from scratch.

files

Displays a list of configuration files in the `/config` file.

Operational mode

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

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

The following example shows how to display the configuration from operational mode. (For brevity, only the first screen of the information is shown.)

```
vyatta@vyatta:~$ show configuration
interfaces {
    dataplane dp0p1p1 {
        address 192.168.1.77/24
        hw-id 00:0c:29:68:b3:9f
    }
    dataplane dp0p1p2 {
        hw-id 00:0c:29:68:b3:a9
    }
    loopback lo {
    }
}
service {
    ssh {
     }
}
system {
    host-name R1
    login {
        user vyatta {
            authentication {
                encrypted-password ****************
:
```

show system commit

Displays a summary of file revisions for a configuration.

```
show system commit
```

Operational mode

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

The following example shows the commit history of system R1.

```
vyatta@R1:~$ show system commit
vyatta@vm-at-1:~$ show system commit
0   2017-01-05 14:49:18 by vyatta
    testing
1   2017-01-05 14:41:32 by vyatta
    Update interface name

After rollback:

vyatta@R1:~$ show system commit
0   2017-01-05 14:50:22 by configd
    Rollback to previous version.
```

```
1   2017-01-05 14:49:18 by vyatta
    testing
2   2017-01-05 14:41:32 by vyatta
    Update interface name
```

show system commit diff

Compares adjacent configuration file revisions.

`show system commit diff` *`rev-num`*

rev-num

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

Operational mode

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

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

The following example shows configuration file revision 18 on R1.

```
vyatta@R1:~$ show system commit diff 18
[edit routing routing-instance red]
-protocols {
-       static {
-               route 20.2.3.0/24 {
-                       next-hop 20.1.2.2
-               }
-               route 20.2.4.0/24 {
-                       next-hop 20.1.2.2
-               }
-               route 20.3.2.0/24 {
-                       next-hop 20.1.2.2
-               }
-               route 20.3.4.0/24 {
-                       next-hop 20.1.2.2
-                       next-hop 20.3.1.2
-               }
-               route 20.4.2.0/24 {
-                       next-hop 20.1.2.2
-               }
-               route 20.4.4.0/24 {
-                       next-hop 20.1.2.2
-               }
-       }
-}
vyatta@R1:~$
```

show system commit file

Displays a specific revision of the configuration file.

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

rev-num

The revision number of the configuration file to display.

rev-num1

The revision number of the configuration file with which to compare.

Operational mode

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

The following example shows revision 0 of the configuration file on R1.

```
vyatta@R1:~$ show system commit file 0
  interfaces {
      dataplane dp0p1p1 {
          address dhcp
          description "bridge to io"
          duplex auto
          speed auto
      }
[... the rest of the configuration file]
vyatta@R1:~$
```

The following example shows two configuration file revisions (18 and 19) being compared on R1.

```
vyatta@R1:~$ show system commit file 18 compare 19
[edit routing routing-instance red]
-protocols {
-       static {
-               route 20.2.3.0/24 {
-                       next-hop 20.1.2.2
-               }
-               route 20.2.4.0/24 {
-                       next-hop 20.1.2.2
-               }
-               route 20.3.2.0/24 {
-                       next-hop 20.1.2.2
-               }
-               route 20.3.4.0/24 {
-                       next-hop 20.1.2.2
-                       next-hop 20.3.1.2
-               }
-               route 20.4.2.0/24 {
-                       next-hop 20.1.2.2
-               }
-               route 20.4.4.0/24 {
-                       next-hop 20.1.2.2
-               }
-       }
-}
[edit]
vyatta@R1:~$
```

system config-management commit-archive location

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

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

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

```
show system config-management commit-archive location
```

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

location

Multinode. A location for the configuration archive. Archives are transferred by any of the following file-transfer methods and their general formats:

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 in which to save the file. The saved file contains the original file name (`config.boot`) followed by the host name of the local system, date (YYYYMMDD), and 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.

Configuration mode

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

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 by using system config-management commit-revisions.

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

However, for this command to succeed with SCP, the router must have the public key of the SCP host. To provide the public key to the router, log in to the SCP host using SSH (SCP uses SSH as its underlying protocol to copy the file) and say 'yes' to the public key that is presented by the SCP host.

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

Use the `delete` form of this command to disable remote archiving of configuration revisions.

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

system config-management commit-revisions

Specifies the number of configuration revisions to store locally.

`set system config-management commit-revisions` *`revisions`*

`delete system config-management commit-revisions`

`show system config-management commit-revisions`

By default, 20 configuration revisions are stored.

revisions

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

Configuration mode

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

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. After the maximum number of revisions has been reached, the oldest revision is removed to make way for a new revision.

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

Use the `set` form of this command to specify the number of locally stored configuration revisions.

Use the `delete` form of this command to restore the default maximum number of 20 revisions.

Use the `show` form of this command to view the maximum number of configuration revisions that are archived locally.

top

Navigates quickly to the top level of the configuration hierarchy.

```
top
```

Configuration mode

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

The following example shows how to navigate down through several nodes of the configuration tree, then use the `top` command to jump directly to the top of the tree. In the example, notice that the [edit] line displays the location in the configuration tree.

```
vyatta@vyatta# edit protocols rip interface dp0p1p1
[edit protocols/rip/interface/dp0p1p1]
vyatta@vyatta# top
vyatta@vyatta#
```

up

Navigates up one level in the configuration hierarchy.

```
up
```

Configuration mode

Use this command to navigate up one level in the configuration hierarchy.

The following example shows how to navigate down through several nodes of the configuration tree, then use the `up` command to navigate successively higher in the tree. In the example, notice that the [edit] line displays the location in the configuration tree.

```
vyatta@vyatta# edit protocols rip interface dp0p1p1
[edit protocols/rip/interface/dp0p1p1]
vyatta@vyatta# up
[edit protocols/rip/interface]
vyatta@vyatta# up
[edit protocols/rip/]
```

Chapter 8. System Management

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

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

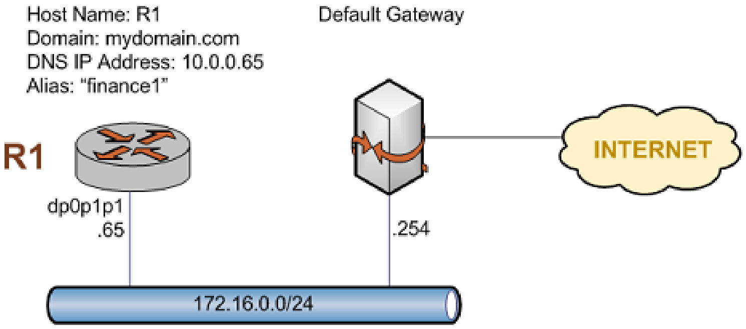
Configuring host information

This section presents the following topics:

- Host name
- Domain
- IP address
- Default gateway
- Aliases

In this section, sample configurations are presented for the host information of the system. The following figure shows the sample information.

Figure 3. Host information



This section includes the following examples:

- Setting the host name of the system
- Setting the domain name of the system
- Mapping the IP address of the system to its hostname
- Setting the default gateway

- Creating an alias for the system

Host name

The name of the router is set by using the `system host-name` command. A system name can include letters, numbers, and hyphens (-).

The following table shows how to set the name of the system to R1. To set the system host name, perform the following steps in configuration mode.

Table 15. Setting the host name of the system

Step	Command
Set the host name of the system.	`vyatta@vyatta# set system host-name R1`
Commit the change. The command prompt changes to reflect the change.	`vyatta@vyatta# commit`

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

Domain

The domain name of the system is set by using the `system domain-name` command. A domain name can include letters, numbers, hyphens (-), and periods (.).

Note: The `system domain-name` and `system domain-search` commands are mutually exclusive. Only one of the two commands can be configured at any one time.

The following table shows how to set the domain name of the system to mydomain.com.

To set the domain name of the system, perform the following steps in configuration mode.

Table 16. Setting the domain name of the system

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 IP address of the system can be statically mapped to its host name for local DNS purposes by 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 a single address, use dotted quad format, that is, *a.b.c.d*. For a network prefix, enter a decimal number from 1 through 32.

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

The following table shows how to create a static mapping between the R1 host name and 10.0.0.65 IP address. The DNS server uses this IP address to resolve DNS requests for R1.mydomain.com.

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

Table 17. Mapping the IP address of the system to its host name

Step	Command
Map the R1 host name to the 10.0.0.65 IP address.	`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

The following table shows how to specify a default gateway for the system at 172.16.0.254.

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

Table 18. Setting the default gateway

Step	Command
Specify the default gateway.	`vyatta@R1# set protocols static route 0.0.0.0/0 next-hop 172.16.0.254`
Commit the change.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show protocols static`
	
	`route 0.0.0.0/0 {`
	`     next-hop 172.16.0.254`
	`}`

Aliases

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

The following table shows how to create the finance1 alias for the system.

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

Table 19. 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`
	`}`

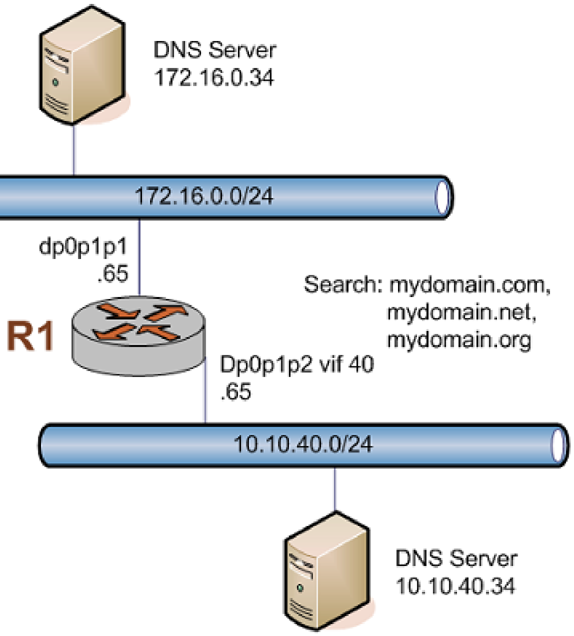
Configuring DNS

This section presents the following topics:

- DNS name servers
- Domain search order

In this section, sample configurations are presented for DNS information. The following figure shows the sample DNS information.

Figure 4. DNS information



DNS name servers

DNS name servers are specified by 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.

The following table shows how to specify two DNS name servers for the system: one at 172.16.0.34 and the other at 10.10.40.34.

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

Table 20. Specifying DNS name servers

Step	Command
Specify the first DNS name server.	`vyatta@R1# set system name-server 172.16.0.34`
Specify the second DNS name 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 by using the `system domain-search` command.

Note: The `system domain-name` and `system domain-search` commands are mutually exclusive. Only one of the two commands can be configured at any one time.

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

The following table shows how to direct the system to attempt domain completion in the following order: first, mydomain.com; second, mydomain.net; and last mydomain.org.

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

Table 21. Specifying the 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`

Table 21. Specifying the search order for domain completion (continued)

Step	Command

	`domain mydomain.net`
	`domain mydomain.org`

Configuring date and time

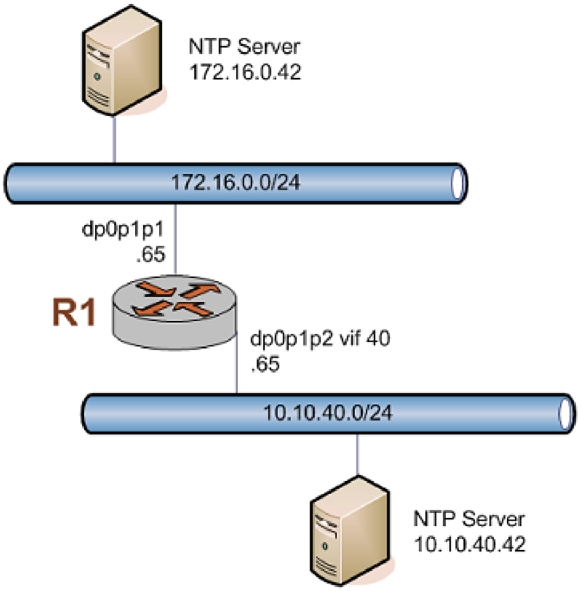
This section presents the following topics:

- Setting the date
- Manually synchronizing with an NTP server
- Setting the time zone
- Using NTP for automatic synchronization

Date and time can be either set manually or obtained by manually or automatically synchronizing the system with one or more Network Time Protocol (NTP) servers. The 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 following figure shows the sample date and time information.

Figure 5. Date and time



Setting the date

The following table shows how to manually set 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.

Table 22. 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

The following table shows how to manually synchronize the system clock with the NTP server at 172.16.0.42.

Note that this action performs just a one-time synchronization. It does not set up an ongoing association with the NTP server. For information about setting up automatic synchronization, refer to Using NTP for automatic synchronization.

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

Table 23. 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

The time zone must be set by using the `system time-zone` command. To set the time zone, you specify the region and location (specified as Region/Location) that best defines your time zone. For example, specifying **US/Pacific** sets the time zone to US Pacific time. Command completion (that is, the <Tab> key) can be used to list available time zones. The adjustment for daylight time takes place automatically based on the time of year.

The following table shows how to set the time zone to Pacific time.

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

Table 24. Setting the time zone as a region and a location

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

The following table shows how to configure 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.

Table 25. 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`

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`
	`    }`

Configuring CPU affinity

By default, the router control and data planes share the CPUs. The data plane uses threads per CPU to perform forwarding work and additional tasks. Its optimization is automatic and high performance for average use environments. However, many environments are not average, for example:

- Extra resources are required for controller or routing protocols.
- Data plane threads must be reduced to avoid consuming resources.
- Interface forwarding threads must be arranged manually for balancing the workload.

The router allows you to configure CPU affinity on the default data plane. CPU affinity allows you to designate a range of CPUs used by the data plane threads.

Note: Misconfiguration of CPU affinity may adversely affect the performance of the router.

If you define CPU affinity for the data plane, the data plane threads are bound to a range of CPUs and executed only on these CPUs. When the CPUs are bound to the data plane, the system and controller threads do not use these CPUs. For example, if the data plane is on an eight CPU system, and the data plane CPU affinity is set to CPUs 1 through 3, then the control and system threads use CPUs 0, and 4 through 7.

Note: The router always allows control threads to run on CPU 0.

To display the number of CPUs available for the data plane, use show hardware cpu.

Configuring CPU affinity on the default data plane

By default, the default data plane use all CPUs. The following example provides the configuration for CPU affinity on the default data plane by using system default dataplane cpu-affinity.

Step	Command

Assign the CPUs for affinity on the default data plane.
Enter the CPU IDs as a range of numbers separated by a hyphen or a comma-separated list.
	`vyatta@vyatta# set system default dataplane`
	` cpu-affinity 1-3,4`

Commit the change.	`vyatta@vyatta# commit`

Show the configuration.	`vyatta@R1# show system default dataplane`
	` default dataplane{`
	`    cpu-affinity 1-3,6`
	` }`

The CPU ID does not have to exist in the system where the data plane is running. For example, if you configure **cpu-affinity** with a range of **0-3** and the data plane is running on a two CPU system, then the data plane only uses CPUs 0 and 1 and silently ignores the other CPUs in the affinity.

If **cpu-affinity** is out of the range of the available CPUs in the data plane environment, for example, if you configure **cpu-affinity** with a range of **4-7** on a two CPU system, the data plane is not started.

Configuring CPU affinity on a data plane interface

By default, the router assigns a CPU to a dataplane interface in a round-robin manner. The following example provides a configuration of CPU affinity on a data plane interface by using system interface dataplane cpu-affinity.

Step	Command
Assign the CPU ID for affinity on a data plane interface. For the CPU ID, enter its number, a range of numbers separated by a hyphen or a comma-separated list.	`vyatta@vyatta# set interface dataplane dp0p192p1 cpu-affinity 1`
Commit the change.	`vyatta@vyatta# commit`

Show the configuration.	`vyatta@R1# show interfaces dataplane`
	`dataplane dp0p192p1 {`
	`    cpu-affinity 1`
	`}`

If you configure a limited number of CPUs, the number of receive queues are reduced. If you configure a CPU that does not exist or is not defined as used for the data plane, then a message is logged and the interface does not start.

Configuring the number of receive queues on a data plane interface

The number of available receive queues on a data interface depends on the NIC type in the router system. In general, the data plane uses two receive queues. However, a 40 GB device may have four queues. Older devices may have one queue. The following example provides the configuration of the receive queues on a data plane interface by using system interface dataplane rx-queues.

Step	Command
Assign two receive queues on a data plane interface. You can enter a number, range separated by a hyphen, or a comma-separated list. You can configure a range if the interface is using multiple queues.	`vyatta@vyatta# set interface dataplane dp0p192p1 rx-queues 1-2`
Commit the change.	`vyatta@vyatta# commit`

Show the configuration.	`vyatta@R1# show interfaces dataplane`
	`dataplane dp0p192p1 {`
	`    rx-queues 1-2`
	`}`

Monitoring system information

This section presents the following topics:

- Showing host information
- Showing the date and time

Showing host information

To view the configured host name, use the `show host name` command in operational mode, as shown in the following example.

```

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

Showing the date and time

To view the date and time according to the system clock, use the `show host date` command in operational mode, as shown in the following example.

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

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

Chapter 9. System Management Commands

Some commands related to certain features of system management are located in other chapters.

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

clear console

Clears the screen of the user console.

```
clear console
```

Operational mode

Use this command to clear the screen of the user console.

clear interfaces counters

Clears interface kernel counters for all interfaces.

```
clear interfaces counters
```

Operational mode

Use this command to clear the kernel counters for all interfaces of all types, including bridge, data plane, loopback and tunnel.

delete session-table

Deletes all entries from the data plane session table.

```
delete session-table
```

Operational mode

Use this command to delete all entries from the data plane session table.

delete session-table conn-id

Deletes all Conntrack entries that match a connection ID from the data plane session table.

```
delete session-table conn-id conn-id
```

conn-id

Conntrack connection ID. The ID ranges from 1 through 4294967296.

Operational mode

Use this command to delete all Conntrack entries that match a connection ID from the data plane session table.

delete session-table destination

Deletes all entries that match the destination IP address of a session from the data plane session table.

```
delete session-table destination destination-ip-address
```

ip-address

Destination IPv4 address. You can specify an IP address (for example, 192.168.1.3) or an IP address and a port (for example, 192.168.1.3:30).

Operational mode

Use this command to delete all entries that match the destination IP address of a session from the data plane session table.

delete session-table destination source

Deletes all entries that match the destination and source IP addresses of a session from the data plane session table.

```
delete session-table destination destination-ip-address source source-ip-address
```

destination-ip-address

Destination IPv4 address. You can specify an IP address (for example, 192.168.1.3) or an IP address and a port (for example, 192.168.1.3:30).

source-ip-address

Source IPv4 address. You can specify an IP address (for example, 192.168.1.3) or an IP address and a port (for example, 192.168.1.3:30).

Operational mode

Use this command to delete all entries that match the destination and source IP addresses of a session from the data plane session table.

delete session-table source

Deletes all entries that match the source IP address of a session from the data plane session table.

```
delete session-table source source-ip-address
```

source-ip-address

Source IPv4 address of of a session. You can specify an IP address (for example, 192.168.1.3) or an IP address and a port (for example, 192.168.1.3:30).

Operational mode

Use this command to delete all entries that match the source IP address of a session from the data plane session table.

delete session-table source destination

Deletes all entries that match the source and destination IP addresses of a session from the data plane session table.

```
delete session-table source source-ip-address destinationdestination-ip-address
```

source-ip-address

Source IPv4 address of a session. You can specify an IP address (for example, 192.168.1.3) or an IP address and a port (for example, 192.168.1.3:30).

destination-ip-address

Destination IPv4 address of a session. You can specify an IP address (for example, 192.168.1.3) or an IP address and a port (for example, 192.168.1.3:30).

Operational mode

Use this command to delete all entries that match the source and destination IP addresses of a session from the data plane session table.

monitor interfaces

Displays bandwidth utilization statistics for each interface across all interfaces.

```
monitor interfaces
```

Operational mode

Use this command to display bandwidth utilization statistics per interface.

Press the question mark (?) key to toggle the following quick reference information:

- Navigation
- Display settings (for example, graphical or detailed statistics)
- Measurement units

The following example shows how to display the bandwidth utilization statistics for each interface on the R1 host.

```
vyatta@R1:~$ monitor interfaces
#   Interface                RX Rate         RX #     TX Rate         TX #
─────────────────────────────────────────────────────
vyatta (source: local)
  0   dp0p5p1                    0.00B            0       0.00B
 0
  1   dp0p5p1.10                 0.00B            0       0.00B
 0
  2   dp0port2                   0.00B            0       0.00B
 0
  3   dp0p2p1                    0.00B            0       0.00B
 0
  4   .spathintf                 0.00B            0       0.00B
 0
  5   lo                         0.00B            0       0.00B
 0
```

poweroff

Powers off the system.

poweroff **[at** *time* **| cancel | now]**

at *time*

The time at which the system is scheduled to be powered off. Set the date, time, or both directly using one of the following formats:

- hh:mm
- MMDDYY
- “hh:mm MMDDYY”
- +mm

Note that the hour field (hh) uses the 24-hour clock (for example, 3:00 PM is represented as 15 in the hour field).

cancel

Cancels a previously scheduled power-off event.

now

Powers off the system without asking for confirmation.

Operational mode

Use this command to power off the system.

Before the system powers off, a message is broadcast to all logged-in users warning them of the power-off event.

Only users with administrative (admin)-level permission can run this command.

The following example shows how to power off the system.

```
vyatta@R1:~$ poweroff
Proceed with poweroff? (Yes/No) [No] y
Broadcast message from root@R1 (tty1) (Mon Dec 17 17:52:37 2012):
The system is going DOWN for system halt NOW!
```

The following example shows how to power off the system at the current time on the specific date of December 11, 2012.

```
vyatta@R1:~$ poweroff at 121112
vyatta@R1:~$
```

The following example shows how to cancel a scheduled power-off event.

```
vyatta@R1:~$ poweroff cancel
vyatta@R1:~$
```

reboot

Reboots the system.

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

at *time*

The time at which the system is scheduled to reboot. Set the date, time, or both directly using one of the following formats:

- hh:mm
- MMDDYY
- “hh:mm MMDDYY”
- **midnight**
- **noon**

Note that the hour field (hh) uses the 24-hour clock (for example, 3:00 PM is represented as 15 in the hour field).

cancel

Cancels a previously scheduled reboot.

now

Reboots the system without asking for confirmation.

Operational mode

Use this command to reboot the system.

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

Only users with administrative (admin)-level permission can run this command.

The following example shows how to reboot the system.

```
vyatta@R1:~$ reboot
Proceed with reboot? (Yes/No) [No] y
Broadcast message from root@R1 (tty1) (Mon Jan 21 17:52:37 2008):
The system is going down for reboot NOW!
```

The following example shows how to reboot the system at the current time on the specific date of December 11, 2009.

```
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
```

The following example shows how to cancel a scheduled reboot.

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

reset ip arp address

Removes entries associated with a specific IP address from the Address Resolution Protocol (ARP) cache.

`reset ip arp address` *`ipv4`*

ipv4

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

Operational mode

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

reset ip arp interface

Removes the entry associated with an Ethernet interface from the Address Resolution Protocol (ADR) cache.

```
reset ip arp interface  interface_name
```

interface_name

The identifier of an interface. Supported interface types are:

- Data plane
- Loopback

For more information about these interface types, refer to Loopback and Data Plane Interfaces.

Operational mode

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

set date

Sets the system date and time directly or specifies a Network Time Protocol (NTP) server from which to acquire them.

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

datetime

The date and time in one of the following formats:

- MMDDhhmm
- MMDDhhmmYY
- MMDDhhmmCCYY
- MMDDhhmmCCYY.ss

Note that the hour field (hh) uses the 24-hour clock (for example, 3:00 PM is represented as 15 in the hour field).

ntpserver

An NTP server from which to acquire the current date and time. You can specify either an IPv4 address or a host name to identify the NTP server.

Operational mode

Use this command to set the system date and time either directly or by specifying an NTP server from which to acquire them. If a time zone has not been configured, then Greenwich mean time (GMT) is assumed. The time zone is set by using system time-zone.

The following example shows how to set the system date and time to May 15, 2008 at 10:55 PM (assuming that the time zone is set to Pacific daylight time).

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

The following example shows how to set the system date and time by using an NTP server at the 69.59.150.135 IP address.

```
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 secvyatta@R1:~$
```

set terminal

Sets the behavior of the system terminal.

`set terminal` { **key query-help** { **enable** | **disable** } | **length** *length* | **pager** [*pager*] | **width** *width* }

key query-help

Enables or disables help by using a question mark (?). The default option is **enable**

length

The number of rows for the display length on the terminal screen.

pager

The program to use as the terminal pager. If no pager is specified, the default (less) is used.

width

The number of columns for the display width on the terminal screen.

Operational mode

Use this command to set the behavior of the system terminal.

show arp

Displays the Address Resolution Protocol (ARP) cache of the system.

`show arp` [*interface*]

interface

An interface for which ARP information is displayed.

Operational mode

Use this command to display the ARP cache of the system.

The following table shows possible ARP states.

Table 26. ARP states

State	Description
Pending	Address resolution is currently being performed on this neighbor entry.
Valid	The neighbor is reachable. Positive confirmation has been received and the path to this neighbor is operational.
Static	This state is a pseudo-state, indicating that this entry should not be cleared from the cache.
Deleted	The arp entry is deleted.
Local	The arp entry is provided on the data plane only

The following example shows how to display the ARP cache of the R1 system.

```
vyatta@R1:~$ show arp
IP Address HW address Dataplane Controller Device
10.18.170.1 0:1b:ed:9f:de:41 VALID VALID dp0p160p1
10.18.170.172 00:0c:29:c6:89:a6 VALID dp0p160p1
vyatta@R1:~$
```

show date

Displays the system date and time in either local time or Universal Time Coordinated (UTC).

`show date [ utc ]`

utc

Displays the date and time in UTC.

Operational mode

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

The following example shows how to display the system date and time on R1.

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

show hardware cpu

Displays CPU information used in the system.

`show hardware cpu [ summary ]`

summary

Shows the CPUs on the system.

Operational mode

Use this command to view CPU information used in the system.

The following example shows CPU information on R1.

```
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 desktop management interface (DMI) of the system.

```
show hardware dmi
```

Operational mode

Use this command to view information about the DMI of the system. The DMI provides a standard framework for managing resources in the device.

The following example shows DMI information on R1.

```
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 memory.

```
show hardware mem
```

Operational mode

Use this command to view information about the system memory.

The following example shows memory information on R1.

```
vyatta@R1:~$ show hardware memory
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 peripheral component interconnect (PCI) bus.

`show hardware pci` [**detailed**]

detailed

Displays detailed information about the PCI bus.

Operational mode

Use this command to view information about the PCI bus. The PCI bus provides communication among the peripheral components and processor of the system.

The following example shows PCI information on R1.

```
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 the command history of the system.

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

The complete command history is displayed.

num

A specific number of recent commands.

brief

Displays the most recent 20 commands.

Operational mode

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

The following example shows history of command execution on R1.

```
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 dataplane dp0p1p1 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 2009-08-05T22:13:58+0000 commit
   21  2009-08-06T05:14:15+0000 show
:
vyatta@R1:~$
```

show host

Displays host information for hosts that can be reached by the system.

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

lookup *hostname*

Shows the canonical name and IP address plus any configured aliases recorded in the name server for the host with the specified host 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 of this system.

os

Shows details about the operating system of the system.

Operational mode

Use this command to view information configured for the host.

The following example shows how to display information about the R2 host.

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

The following example shows how to display the name of the R1 host.

```

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

The following example shows how to display the date and time according to the system clock.

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

The following example shows how to display information about the host operating system.

```
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 ip groups

Displays IP groups status.

`show ip groups`

Operational mode

Use this command to display IP groups status.

The following example shows how to display the status of IP forwarding.

```
vyatta@vyatta:~$ show ip groups
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 0 127.0.0.1:5904 127.0.0.1:49746 ESTABLISHED
tcp 0 0 10.1.17.201:22 10.250.0.79:58689 ESTABLISHED
tcp 0 0 127.0.0.1:49746 127.0.0.1:5904 ESTABLISHED
tcp 0 0 127.0.0.1:5903 127.0.0.1:48562 ESTABLISHED
tcp 0 0 127.0.0.1:48562 127.0.0.1:5903 ESTABLISHED
IPv6/IPv4 Group Memberships
Interface RefCnt Group
--------------- ------ ---------------------
lo 1 224.0.0.1
dp0p160p1 1 224.0.0.1
.spathintf 1 224.0.0.1
```

show interfaces

Displays information about system interfaces.

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

Displays information for all interfaces configured on the system.

counters

Displays kernel counter information about all the interfaces available on your system.

detail

Displays detailed information about all the interfaces available on your system.

system

Displays all the physical interfaces available on your system.

enabled

Displays only enabled system interfaces known to the operating system kernel.

Operational mode

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

When used with no option, this command 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 are able to configure and view because you cannot configure or view an interface that does not physically exist on the system.

The following example shows how to display information for interfaces.

```
vyatta@R1:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
dp0p2p1          192.168.122.30/24                 u/u
dp0p5p1          -                                 u/u
dp0p5p1.10       10.1.1.1/24                       u/u
dp0port2         -                                 A/D
lo               127.0.0.1/8                       u/u
                 ::1/128
```

The following example shows how to display detailed information for interfaces.

```
vyatta@R1:~$ show interfaces system enabled
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
 DEFAULT group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00 promiscuity 0
    RX: bytes  packets  errors  dropped overrun mcast
    70108352   432856   0       0       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    70108352   432856   0       0       0       0
6: dp0p160p1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
 state UP mode DORMANT group default qlen 500
    link/ether 00:0c:29:2e:2a:d7 brd ff:ff:ff:ff:ff:ff promiscuity 0
    tun
    RX: bytes  packets  errors  dropped overrun mcast
    38258251   588550   0       190834  0       566086
    TX: bytes  packets  errors  dropped carrier collsns
    982191     11700    0       0       0       0
7: dp0p192p1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
 state UP mode DORMANT group default qlen 500
    link/ether 00:0c:29:2e:2a:e1 brd ff:ff:ff:ff:ff:ff promiscuity 0
    tun
    RX: bytes  packets  errors  dropped overrun mcast
    120        2        0       3       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    110        1        0       0       0       0
8: dp0p224p1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
 state UP mode DORMANT group default qlen 500
    link/ether 00:0c:29:2e:2a:eb brd ff:ff:ff:ff:ff:ff promiscuity 0
    tun
    RX: bytes  packets  errors  dropped overrun mcast
    120        2        0       0       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    408        4        0       0       0       0
```

```
10: .spathintf: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
 state UNKNOWN mode DEFAULT group default qlen 500
    link/ether 72:09:8f:fd:1e:38 brd ff:ff:ff:ff:ff:ff promiscuity 0
    tun
    RX: bytes  packets  errors  dropped overrun mcast
    0          0        0       0       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    408        4        0       0       0       0
```

show interfaces extensive

Displays detailed information about all system interfaces.

```
show interfaces extensive
```

Operational mode

Use this command to view detailed configuration information and operational status for all interfaces and virtual interfaces. In the output of this command where an interval is referenced, the average values are 1-minute-, 5-minute-, and 15-minute-weighted rolling average values computed in a manner similar to the values reported by Unix and Linux for load average in the uptime command.

Note:

For better-formatted output and more complete information about an interface, use the **show interfaces dataplane** *interface-name* command.

The following example shows how to display detailed information about all system interfaces.

```
vyatta@R1:~$ show interfaces extensive
dp0p192p1:
  rx bad address: 0                        rx bad vid: 0

  rx badcrc: 0                             rx badlen: 0

  rx bps: 24                               rx bps avg: 62, 82, 83

  rx bridge: 0                             rx bytes: 132549906

  rx dropped: 0                            rx error: 0

  rx errors: 0                             rx flowmatch: 0

  rx flowmiss: 0                           rx good bytes: 132549906

  rx good packets: 1246364                 rx mbuf allocation errors: 0

  rx missed: 0                             rx multicast: 1225784
  rx nobuffer: 0                          rx non ip: 1081511

  rx packets: 1246364                     rx pps: 0

  rx pps avg: 0, 0, 0                     rx q0 bytes: 130658262

  rx q0 errors: 0                         rx q0 packets: 1225457

  rx q1 bytes: 1891644                    rx q1 errors: 0

  rx q1 packets: 20907                    rx vlan: 0


  tx bps: 0                               tx bps avg: 5, 12, 9

  tx bytes: 1941172                       tx dropped: 0

  tx error: 0                             tx errors: 0

  tx good bytes: 1941172                  tx good packets: 18897

  tx packets: 18897                       tx pps: 0

  tx pps avg: 0, 0, 0                     tx q0 bytes: 1928872

  tx q0 packets: 18692                    tx q1 bytes: 0

  tx q1 packets: 0                        tx q2 bytes: 12300

  tx q2 packets: 205                      tx q3 bytes: 0

  tx q3 packets: 0

dp0p224p1:
  rx bad address: 0                       rx bad vid: 0

  rx badcrc: 0                            rx badlen: 0

  rx bps: 0                               rx bps avg: 0, 0, 0

  rx bridge: 0                            rx bytes: 120

  rx dropped: 0                           rx error: 0

  rx errors: 0                            rx flowmatch: 0
...
```

show license

Displays Vyatta license information.

```
show license
```

Operational mode

Use this command to view Vyatta license information.

The following example shows how to display Vyatta 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 connections to a configured Network Time Protocol (NTP) server.

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

host

The host name of an NTP server.

ipv4

The IPv4 address of an NTP server.

0.vyatta.pool.ntp.org

Specifies the default NTP server.

Operational mode

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

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

NTP server connections are configured by using system ntp server.

The following example shows how to display the status of connections to all configured NTP servers (in this case, the 69.59.150.135 IP address).

```
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:~$
```

The following example shows how to display the status of connections to the configured NTP server at the 69.59.150.135 IP address.

```
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 ntp packets

Displays the number of NTP packets sent and received. It also displays counts of packets that caused exceptional conditions.

`show ntp` **`packets`**

Operational mode

Use this command to display the number of NTP packets sent and received and counts of packets that caused various exceptional conditions.

The following example shows how to display the number of NTP packets sent and received and counts of packets that caused exceptional conditions.

```
vyatta@R1:~$ show ntp packets
    packets sent:            57
    packets not sent:        0
    packets received:        59
    packets processed:       54
    current version:         54
    previous version:        0
    declined:                0
    access denied:           0
    bad length or format:    0
    bad authentication:      0
rate exceeded:           0
```

show ntp status

Displays an overview of the NTP daemon and the peer to which the NTP server is synchronizing.

`show ntp` **`status`**

Operational mode

Use this command to display an overview of the NTP daemon and the peer to which the NTP server is synchronizing.

The following example shows how to display an overview of the NTP daemon.

```

vyatta@R1:~$ show ntp status
     system peer:         64.246.132.14
    system peer mode:     client
    leap indicator:       11
    stratum:              2
    precision:            -23
    root distance:        0.12462 s
    root dispersion:      0.07733 s
    reference ID:         [64.246.132.14]
    reference time:       d8592b62.8bdfaa14  Thu, Jan  8 2015 16:14:26.546
    system flags:         auth monitor ntp kernel stats
    jitter:               0.003525 s

    System clock is synchronized
```

show ntp information

Displays version information for the NTP daemon and indicates if the process is running.

`show ntp` **`information`**

Operational mode

Use this command to display version information for the NTP daemon and to check whether the process is running.

The following example shows how to display version information for the NTP daemon and check whether the process is running.

```
vyatta@R1:~$ show ntp information
version 1:4.2.6.p5+dfsg-2+deb7u1+vyatta1+1420850908
NTP daemon is running
```

show session-table

Shows the dataplane session table details.

show session-table [**source** *src-addr* | **statistics**]

src-address

Source IP address. Filters the output based on either a source IP address or a source IP address combined with a port.

Operational mode

Use this command to view information on the dataplane session table.

The following example shows how to display information about the complete session table.

```
vyatta@vyatta% show session-table
TCP state codes: SS - SYN SENT, SR - SYN RECEIVED, ES - ESTABLISHED,
                 FW - FIN WAIT, CW - CLOSE WAIT, LA - LAST ACK,
                 TW - TIME WAIT, CL - CLOSE, LI - LISTEN

CONN ID         Source                          Destination
     Protocol        TIMEOUT Intf            Parent
22              10.0.1.1:2601                   10.0.2.1:100
     tcp [6] SS      25      dp0s11  0
23              10.0.1.1:2602                   10.0.2.1:100
     tcp [6] SS      26      dp0s11  0
24              10.0.1.2:2603                   10.0.2.1:100
     tcp [6] SS      27      dp0s11  0
25              10.0.1.2:2604                   10.0.2.1:100
     tcp [6] SS      28      dp0s11  0
```

The following example shows how to display details of the session table filtered by source address.

```
vyatta@vyatta% show session-table source 10.0.1.2
```

```
TCP state codes: SS - SYN SENT, SR - SYN RECEIVED, ES - ESTABLISHED,
                 FW - FIN WAIT, CW - CLOSE WAIT, LA - LAST ACK,
                 TW - TIME WAIT, CL - CLOSE, LI - LISTEN

CONN ID         Source                          Destination
     Protocol        TIMEOUT Intf            Parent
24              10.0.1.2:2603                   10.0.2.1:100
     tcp [6] SS      27      dp0s11  0
25              10.0.1.2:2604                   10.0.2.1:100
     tcp [6] SS      28      dp0s11  0
```

The following example shows how to display details of the session table filtered by source address and port.

```
vyatta@vyatta% show session-table source 10.0.1.1:2602
TCP state codes: SS - SYN SENT, SR - SYN RECEIVED, ES - ESTABLISHED,
                 FW - FIN WAIT, CW - CLOSE WAIT, LA - LAST ACK,
                 TW - TIME WAIT, CL - CLOSE, LI - LISTEN

CONN ID         Source                          Destination
     Protocol        TIMEOUT Intf            Parent
23              10.0.1.1:2602                   10.0.2.1:100
     tcp [6] SS      26      dp0s11  0
```

The following example shows how to display a summary of the session table statistics.

```
vyatta@vyatta% show session-table statistics
Available (percentage): 984064 (93.85%)
Used (percentage): 64512 (6.15%)
NATed: 64512
Detailed (by state):
TCP
SYN SENT: 0
SYN RECEIVED: 0
ESTABLISHED: 0
FIN WAIT: 0
CLOSE WAIT: 0
LAST ACK: 0
TIME WAIT: 0
CLOSE: 0
LISTEN: 0
UDP
NEW: 0
ESTABLISHED: 64512
CLOSE: 0
Other
NEW: 0
ESTABLISHED: 0
CLOSE: 0
```

show reboot

Shows the next scheduled reboot date and time.

```
show reboot
```

Operational mode

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

The following example shows how to display the next scheduled reboot date and time.

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

The following example shows that no reboot is scheduled.

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

show system boot-messages

Displays bootup messages generated by the kernel.

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

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

all

Displays all kernel bootup messages.

Operational mode

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

The following example shows how to display bootup messages that have been generated by the kernel.

```
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.

`show system connections`

Operational mode

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

The following example shows how to display active network connections on the system.

```
vyatta@R1:~$ show system connections
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:5903          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:5904          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:5907          0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:53              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:5904          127.0.0.1:42165
 ESTABLISHED
tcp        0      0 127.0.0.1:42165         127.0.0.1:5904
 ESTABLISHED
tcp        0      0 127.0.0.1:48564         127.0.0.1:5903
 ESTABLISHED
tcp        0     64 10.1.17.201:22          10.250.1.136:61388
 ESTABLISHED
tcp        0      0 127.0.0.1:5903          127.0.0.1:48564
 ESTABLISHED
```

```
tcp6       0      0 :::53                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
udp        0      0 0.0.0.0:53              0.0.0.0:*
udp        0      0 10.1.17.201:123         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 :::53                   :::*
udp6       0      0 fe80::ff:fe00:1:123     :::*
udp6       0      0 fe80::250:56ff:fea9:123 :::*
udp6       0      0 ::1:123                 :::*
udp6       0      0 :::123                  :::*
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     9478
  /var/run/vplane.socket
unix  2      [ ACC ]     STREAM     LISTENING     6702
  /var/run/vcfgfs.sock
unix  2      [ ACC ]     STREAM     LISTENING     14164
  /tmp/browser_pager
unix  2      [ ACC ]     STREAM     LISTENING     7765     /tmp/.rip_show
unix  2      [ ACC ]     STREAM     LISTENING     8796
  /var/run/vyatta/vplaned.socket
unix  2      [ ACC ]     STREAM     LISTENING     7772     /tmp/.ripng_show
unix  2      [ ACC ]     STREAM     LISTENING     7779     /tmp/.ospf_show
unix  2      [ ACC ]     STREAM     LISTENING     7786     /tmp/.ospf6_show
unix  2      [ ACC ]     STREAM     LISTENING     6021     /tmp/.bgp_show
unix  2      [ ACC ]     STREAM     LISTENING     6793     /tmp/.imi_show
unix  2      [ ACC ]     STREAM     LISTENING     6797     /tmp/.imi_line
unix  2      [ ACC ]     STREAM     LISTENING     6811
  /var/run/acpid.socket
unix  2      [ ACC ]     STREAM     LISTENING     8603     /tmp/.nsm_show
unix  2      [ ACC ]     STREAM     LISTENING     8607     /tmp/.nsmserv
<omitted>
```

show system kernel-messages

Displays messages in the kernel ring buffer.

```
show system kernel-messages
```

Operational mode

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

The following example shows how to display messages in the kernel ring buffer.

```

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.

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

cache

Displays memory cache details.

detail

Displays memory usage details.

Operational mode

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

The following example shows information about memory usage on R1.

```
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:~$
```

The following example shows detailed information about memory usage on R1.

```
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
:
```

The following example shows information about memory cache usage on R1.

```
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 power-profile

Displays the current power profile settings.

`show system power-profile`

Configuration mode

Use the `show` form of this command to display the power profile settings.

Displaying power profile settings

The following example shows how to display the power profile settings.

```

vyatta@R1# run show system power-profile
balanced (100, 10, 250)
```

show system processes

Displays information about processes currently running on the system.

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

Lists all processes currently running on the system.

extensive

Shows all processes and extensive details about each process.

summary

Shows a summary of system usage.

tree

Shows all processes and how they are related.

Operational mode

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

The following example shows how to display a list of all processes currently running on the system.

```

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]
:
```

The following example shows how to display extensive information about all processes currently running on the system.

```
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
```

```
:
```

The following example shows how to display all processes that are currently running and how they are related.

```
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 a list of active routing daemons.

```
show system routing-daemons
```

Operational mode

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

The following example shows how to display a list of active routing daemons.

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

show system storage

Displays system file usage and available storage space.

```
show system storage
```

Operational mode

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

The following example shows file system usage information for R1.

```
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.

```
show system uptime
```

Operational mode

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

The following example shows file system usage information for R1.

```
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 which peripherals are connected to the USB bus.

```
show system usb
```

Operational mode

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

The following example shows system USB information for R1.

```
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

Displays a consolidated report of system information.

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

Information is sent to the console.

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 limits the number of output files to 100; that is, creating a file after the first 100 files 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 limits the number of output files to 100; that is, creating a file after the first 100 files causes the oldest file to be deleted.

filename

The name of a file to which to save the support information. Refer to “Usage Guidelines” for details.

Operational mode

Use this command to display a technical report that provides consolidated information about system components and configuration.

Note: Only administrative (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 IP Infusion Inc. technical support.

Technical 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 technical support directory is `/config/support`.

If a file name is specified, the support information is saved to the **filename**.*hostname*.**tech-support**.*timestamp* file, 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 prefixed to the file name, the file is saved in that location. Otherwise, the file is saved to a location relative to the default path, which is the `/config/support` directory. An FTP or SCP server can also be specified.

The following table shows how to specify the syntax for files from different file locations.

Table 27. 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.

The following example shows how to display a technical report of 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 {
    dataplane dp0p1p1 {
        address 192.168.1.82/24
        duplex auto
:
```

show version

Displays information about the versions of system software.

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

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

all

Displays all software that has been added, deleted, downgraded, or upgraded since the last baseline version upgrade.

added

Displays all packages that have been added since the last baseline version upgrade.

deleted

Displays all packages that have been deleted since the last baseline version upgrade.

downgraded

Displays all packages that have been downgraded since the last baseline version upgrade.

upgraded

Displays all packages that have been upgraded since the last baseline version upgrade.

Operational mode

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

The information shown always relates to the last full version upgrade. Therefore, the following conditions apply.

- Immediately after a full version upgrade, entering a `show version all` command displays no changes.

- If a package is added after an upgrade, entering a `show version all` command displays the added package.
- However, if the added package is then deleted again, entering a `show version all` command displays no change because the system is now in the same state as it is immediately after the full version upgrade.

If there are no added, deleted, upgraded, or downgraded packages the output does not provide any package information.

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

If there have been packages added, deleted, upgraded and/or downgraded the package information will immediately follow the general version information (see above). The output will list the added packages first, followed by all the deleted packages, followed by the list of upgraded packages, and finally ending with a list of downgraded packages.

The following example shows a version summary.

```
vyatta@vyatta:~$ show version
Version:      2005
Description:  DANOS - Vyatta NOS Edition 2005 DVE
Built on:     Fri Jul 24 14:49:42 UTC 2020
System type:  Intel 64bit
Boot via:     image
HW model:     S9500-30XS
HW S/N:       WCF1997F00297
HW UUID:      03000200-0400-0500-0006-000700080009
Uptime:       16:56:54 up 10:59,  1 user,  load average: 0.63, 0.52, 0.62
```

system alg ftp

Configures tracking of FTP connections.

`set system alg ftp {` **`disable`** `|` **`port`** *`port-number`* `}`

`delete system alg ftp` **`port`** *`port-number`*

`show system alg ftp` **`port`** *`port-number`*

FTP connection tracking is enabled.

disable

Disables tracking of FTP connections.

port *port-number*

Specifies a control port for the tracking of FTP connections.

Configuration mode

```
system {
    alg{
        ftp {
            disable
            port port-number
        }
    }
}
```

Use the `set` form of this command to configure tracking of FTP connections.

Use the `delete` form of this command to remove a port from the tracking of FTP connections.

Use the `show` form of this command to display the configuration of FTP connection tracking.

system alg icmp disable

Disables tracking of ICMP connections.

```
set system alg icmp  disable
```

```
delete system alg icmp  disable
```

```
show system alg icmp  disable
```

ICMP connection tracking is enabled.

Configuration mode

```
system {
   alg{
      icmp {
         disable
      }
   }
}
```

ALGs work as helpers for the NAT system for a specified protocol. ALGs are enabled by default when NAT is enabled. Disabling an ALG may result in NAT not being performed correctly for the specified protocol.

Use the `set` form of this command to disable ICMP connection tracking.

Use the `delete` form of this command to reenable ICMP connection tracking.

Use the `show` form of this command to display the configuration of ICMP connection tracking.

system console device

Defines a specified device as the system console.

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

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

```
show system console device  device
```

The serial port device (**ttyS0**) is configured with a speed of **115200** .

device

Multi-node. The name of a console device. The device name is one of the following:

ttySN: Serial device name

ttyUSBX: USB serial device name

hvc0: Xen console

ttyS0: Serial port device

speed

The speed (baud rate) of the console device. The speed is one of the following: **1200**, **2400**, **4800**, **9600**, **19200**, **38400**, **57600**, or**115200.** The default speed is **115200 .**

modem

Indicates that the port is connected to the serial console through a Hayes compatible modem.

Configuration mode

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

Use this command to specify a device as the system console.

Changes take effect the next time a user logs in through the device and not when the configuration is committed.

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

Bootup messages are limited to the serial port device (ttyS0). Other consoles can be configured but do 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 a device as the system 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 network

Sends console messages to a remote system.

```
set system console network  device [ interface ethx ] [ local [ address ipv4 ] [
port port ] ] [ remote [ address ipv4 ] [ mac macaddr ] [ port port ] ]
```

```
delete system console network  device [ interface ] [ local [ address ] [ port ]
] [ remote [ address ] [ mac ] [ port ] ]
```

```
show system console network  device [ interface ] [ local [ address ] [ port ] ]
[ remote [ address ] [ mac ] [ port ] ]
```

Console messages are sent as broadcasts if no remote address is specified.

device

Multi-node. The name of a remote console device (for example, `netcon00`).

ethx

Mandatory. An Ethernet interface on which to send out messages.

local address *ipv4*

Specifies an IPv4 address to use as the source address. The address must be configured on the interface specified.

local port *port*

Specifies a port by its port number to use as the source port. The port number ranges from 1 through 65535. The default number is 6665.

remote address *ipv4*

Specifies the IPv4 address of the destination system.

remote mac *macaddr*

Specifies the MAC address of the destination system.

remote port *port*

Specifies a port by its port number to use as the destination port. The port number ranges from 1 through 65535. The default number is 6666.

Configuration mode

```
system {
    console {
    network device {
        interface ethx
        local  {
               address ipv4
               port port
               }
        remote {
               address ipv4
               mac macaddr
               port port
        }
      }
   }
}
```

Use this command to send console messages to a remote system over UDP.

Use the `set` form of this command to create a network console.

Use the `delete` form of this command to remove a network console.

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

system console powersave

Saves power when a blank screen appears on the VGA console.

```
set system console powersave
```

```
delete system console powersave
```

```
show system console
```

Power is not saved.

Configuration mode

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

Use this command to save power when a blank screen appears on the VGA console. After 15 minutes of inactivity the screen goes blank. After 60 minutes, the monitor powers down.

Use the `set` form of this command to save power when a blank screen appears on the console.

Use the `delete` form of this command to return the system to its default behavior, that is, power is not saved.

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

system default dataplane cpu-affinity

Assigns a range or list of CPUs to the affinity on the default data plane.

`set system default dataplane` **`cpu-affinity`** *`cpu-list`*

`delete system default dataplane [` **`cpu-affinity`** `[` *`cpu-list`* `] ]`

`show system default dataplane [` **`cpu-affinity`** `]`

cpu-list

The CPU IDs assigned to the affinity. Enter a range of numbers separated by a hyphen or a comma-separated list. The first CPU is 0. At least two CPUs must be assigned.

Configuration mode

```
system {
    default dataplane {
        cpu-affinity cpu-list
    }
}
```

Note: Misconfiguration of CPU affinity may adversely affect the performance of the router.

By default, all CPUs are used.

To display the number of CPUs available for the data plane, use show hardware cpu.

The CPU ID does not have to exist in the system where the data plane is running. For example, if you configure **cpu-affinity** with a range of **0-3** and the data plane is running on a two CPU system, then the data plane only uses CPUs 0 and 1 and silently ignores the other CPUs in the affinity.

If **cpu-affinity** is out of the range of the available CPUs in the data plane environment, for example, if you configure **cpu-affinity** with a range of **4-7** on a two CPU system, then an ERROR priority system message is logged and the data plane is not started.

Use the `set` form of this command to define the CPU affinity list for the default data plane.

Use the `delete` form of this command to remove CPU affinity list for the default data plane.

Use the `show` form of this command to view the CPU affinity list for the default data plane.

system domain-name

Establishes a domain name for the system.

`set system domain-name` *`domain`*

`delete system domain-name`

```
show system domain-name
```

domain

Mandatory. A name for the domain in which the system resides. The format of the name is a character string that contains letters, numbers, hyphens (-), and one period; for example, ipinfusion.com. A domain name can have a maximum of 253 characters.

Configuration mode

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

Use this command to establish a domain name for the system.

Note that both the `system domain-name` and `system domain-search domain` commands cannot be configured simultaneously; they are mutually exclusive.

Use the `set` form of this command to establish the domain name to be used by the system.

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

Use the `show` form of this command to display a domain name.

system domain-search domain

Defines a set of domains for domain completion.

```
set system domain-search domain  domain
```

```
delete system domain-search domain  domain
```

```
show system domain-search domain
```

domain

Mandatory. Multi-node. A domain name to be added to or deleted from the list of domains in the search order string. The format of the name is a character string that contains letters, numbers, hyphens (-), and one period; for example, ipinfusion.com. A domain name can have a maximum of 253 characters.

You can specify up to six domains by creating up to six **domain-search** multi-nodes.

Configuration mode

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

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 the `system domain-name` and `system domain-search domain` commands cannot be configured simultaneously; they are mutually exclusive.

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

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

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

system host-name

Establishes the host name for the system.

```
set system host-name  name
```

```
delete system host-name
```

```
show system host-name
```

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

name

A name you want to give to the system. The name can contain only letters, numbers, and hyphens (-).

The default name is **vyatta**. If you delete the host name, or if you try to delete the **system** node, the host name reverts to the default name of **vyatta**.

Configuration mode

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

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

When you establish the name, 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 establish or change the host name.

Use the `delete` form of this command to restore the default host name of **vyatta**.

Use the `show` form of this command to display the host name.

system interface dataplane cpu-affinity

Assigns CPU affinity to a data plane interface.

`set system interface dataplane` *id* **cpu-affinity** *cpu-list*

`delete system interface dataplane` *id* [**cpu-affinity** [*cpu-list*]]

`show system interface dataplane` *id* [**cpu-affinity**]

id

The data plane ID.

cpu-list

The CPUs assigned to the affinity. Enter a number, a range of numbers separated by a hyphen or a comma-separated list.

Configuration mode

```
system {
    interface dataplane id {
        cpu-affinity cpu-list
    }
}
```

By default, the router assigns a CPU to a dataplane interface in a round-robin manner.

If you configure a limited number of CPUs, the number of receive queues are reduced. If you configure a CPU that does not exist or is not defined as used for the data plane, then a message is logged and the interface does not start.

Use the `set` form of this command to define the CPU affinity list for the data plane interface.

Use the `delete` form of this command to remove CPU affinity list for the data plane interface.

Use the `show` form of this command to view the CPU affinity list for the data plane.

system interface dataplane rx-queues

Assigns the number of receive queues for a data plane interface.

`set system interface dataplane` *id* **rx-queues** *number*

`delete system interface dataplane` *id* [**rx-queues** [*number*]]

`show system interface dataplane` *id* [**rx-queues**]

id

The data plane interface ID.

number

The number of receive queues for the interface. Enter a number, range separated by a hyphen, or a comma-separated list. You can configure a range if the interface is using multiple queues.

Configuration mode

```
system {
    interface dataplane id {
        rx-queues number
    }
}
```

The number of available receive queues on a data interface depends on the NIC type in the router system. In general, the data plane uses two receive queues. However, a 40 GB device may have four queues. Older devices may have one queue.

Use the `set` form of this command to define the number of receive queues on the data plane interface.

Use the `delete` form of this command to remove the number of configured receive queues for the data plane interface

Use the `show` form of this command to view the number of receive queues for the data plane interface.

system name-server

Specifies a Domain Name System (DNS) name server for the system.

```
set system name-server  address
```

```
delete system name-server  address
```

```
show system name-server
```

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.

Configuration mode

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

Use this command to a DNS for the system.

Use the `set` form of this command to specify a name server for the system. Note that you cannot modify the entry of a DNS name server by using the `set` command. To replace an entry, delete it and create a new entry.

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 specified.

system ntp server

Specifies a Network Time Protocol (NTP) server to use when synchronizing the system clock.

`set system ntp server` *server* [**address-family** | **dynamic** | **keyid** | **noselect** | **preempt** | **prefer**]

`delete system ntp server` *server* [**address-family** | **dynamic** | **keyid** | **noselect** | **preempt** | **prefer**]

`show system ntp server`

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

server

Multi-node. The IP address or host name of an NTP server. The system automatically obtains the system date and time from the specified server or servers.

You can specify multiple NTP servers by creating multiple instances of the ntp server configuration node.

address-family

Address family for hostname resolution.

dynamic

Allows to configure the server even if it is not reachable.

key-id

NTP aunthentication key ID.

noselect

Marks the server as unused.

preempt

Specifies the association as preemptable rather than the default persistent.

prefer

Marks the server as preferred.

Configuration mode

```
system {
    ntp {
       server server {
          address-family
          dynamic
          key-id
          noselect
          preempt
          prefer
       }
    }
}
```

Use this command to specify an NTP server 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 by using the `set` command. To replace an entry, delete it and create a new entry.

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 specified.

This example describes mark a NTP server 10.18.191.203 as the preferred server.

```
vyatta@Rn# set system ntp server 10.18.191.203 prefer
vyatta@Rn# show system ntp server
server 10.18.191.203 {
       prefer
```

system power-profile policy

Creates the idle, minimum sleep time, and maximum sleep time thresholds for a power profile.

`set system power-profile policy` [**balanced** | **power-save** | **low-latency**]

`delete system power-profile policy` [**balanced** | **power-save** | **low-latency**]

`show system power-profile policy` [**balanced** | **power-save** | **low-latency**]

The default setting is `balanced`.

balanced

Provides the best overall performance, but adds latency to the handling of the initial packet in a burst.

power-save

The polling parameters are adjusted to optimize the utilization of the CPU, but adds latency to the initial packet.

low-latency

The polling parameters are adjusted to optimize for low packet latency at the expense of CPU utilization.

Configuration mode

This command allows administrators to adjust the idle threshold, minimum sleep time, and maximum sleep time.

The data plane determines how long the CPU core sleeps between polls for packets based on how busy the CPU core has been. When the CPU sees multiple packets when polling a device, it considers itself busy and cuts the sleep time in half. If the CPU has not seen any packets in the given interval, it considers itself idle and increases the sleep interval by one microsecond.

Use the `set` form of this command to create policy settings for a power profile.

Use the `delete` form of this command to delete the policy settings for a power profile.

Use the `show` form of this command to show the policy settings for a power profile.

system power-profile custom

Creates the thresholds for a custom policy of a power profile.

```
set system power-profile custom  parameter [ idle-threshold microseconds | min-sleep microseconds | max-sleep microseconds ]
```

```
delete system power-profile custom  parameter [ idle-threshold | min-sleep | max-sleep ]
```

```
show system power-profile custom  parameter [ idle-threshold | min-sleep | max-sleep ]
```

idle-threshold *microseconds*

Sets the idle threshold in microseconds.

min-sleep *microseconds*

Sets the minimum sleep time in microseconds.

max-sleep *microseconds*

Sets the maximum sleep time in microseconds.

Configuration mode

Use the `set` form of this command to create thresholds for a custom policy of a power profile.

Use the `delete` form of this command to delete the thresholds for a custom policy of a power profile.

Use the `show` form of this command to show the thresholds for a custom policy of a power profile.

system ntp server address-family

Specifies the address family for a Network Time Protocol (NTP) server.

`set system ntp server` *server* **address-family** [**ipv4** | **ipv6**]

`delete system ntp server` *server* **address-family** [**ipv4** | **ipv6**]

`show system ntp server`

By default, the system uses the NTP server at 0.vyatta.pool.ntp.org. If no address family is specified then the address selection is determined by the resolver.

server

Multi-node. The IP address or host name of an NTP server. The system automatically obtains the system date and time from the specified server or servers.

You can specify multiple NTP servers by creating multiple instances of the ntp server configuration node.

Configuration mode

```
system {
    ntp {
       server server
         address-family ipv4
         address-family ipv6
    }
}
```

Use this command to specify the address family for an NTP server.

When specified, the address-family parameter forces the name resolution to choose an IP address within that family. This is useful when both A and AAAA records exist in DNS for the same host name.

Note: Time servers in the second address pool, 2.vyatta.pool.ntp.org, have IPv6 connectivity.

Use the `set` form of this command to specify the address family for a NTP server.

You cannot modify an NTP server entry by using the `set` command; to replace an entry, delete it and create a new entry.

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 specified.

system options reboot-on-panic

Specifies whether to reboot the system if a kernel panic occurs.

```
set system options reboot-on-panic  value
```

```
delete system options reboot-on-panic
```

```
show system options reboot-on-panic
```

The system reboots (**true**).

value

Mandatory. Indicates whether the system should automatically reboot if a kernel panic occurs. The value is one of the following:

true—Reboots the system

false—Does not reboot the system

Configuration mode

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

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 to reboot the system if a kernel panic occurs.

Use the `delete` form of this command to restore default behavior, that is, the system reboots.

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

system session table-size

Sets the maximum size of the connection-tracking table.

```
set system session table-size  number
```

```
delete system session table-size
```

```
show system session table-size
```

1,048,576

number

The maximum number of entries allowed in the connection-tracking table. The number ranges from 1 to 100000000.

Configuration mode

```
system {
   session {
      table-size number {
      }
   }
}
```

Use the `set` form of this command to set the maximum size of the connection-tracking table.

Use the `delete` form of this command to restore the default size of the connection-tracking table.

Use the `show` form of this command to display the table size.

When you configure connection synchronization on a router, the maximum number of session entries that you can configure is 200000 when the system memory is 4G, or 100000 entries when the system memory is 2G.

system session timeout custom rule rule-number destination

Specifies destination parameters for custom session timeout.

```
set system session timeout custom rule   rule-number destination { address
address | port port-number }
```

```
delete system session timeout custom rule   rule-number
```

```
show system session timeout custom rule  rule-number
```

rule-number

Number to identify the custom rule. The rules are evaluated in numeric order. The first match is used to apply the timeout value.

address

Destination IP address, subnet, or address group.

port

Destination port or port group.

Configuration mode

```
system {
    session {
       timeout {
          custom {
            rule rule-number {
               destination {
                 address address
                 port port
                 }
             }
          }
       }
   }
}
```

Use this command to specify custom destination parameters for session timeouts.

Use the `set` form of this command to specify custom parameters.

Use the `delete` form of this command to remove custom settings.

Use the `show` form of this command to view the current settings.

system session timeout custom rule rule-number expire

Specifies custom session expiration time.

```
set system session timeout custom rule   rule-number expire time
```

```
delete system session timeout custom rule   rule-number
```

```
show system session timeout custom rule  rule-number
```

rule-number

Number to identify the custom rule. The rules are evaluated in numeric order. The first match is used to apply the timeout value.

time

Interval after which the session expires (1-21474836 seconds).

Configuration mode

```
system {
    session {
       timeout {
          custom {
            rule rule-number {
                 expire time
             }
          }
       }
   }
}
```

Use this command to specify custom session expiration time.

Use the `set` form of this command to specify custom parameters.

Use the `delete` form of this command to remove custom settings.

Use the `show` form of this command to view the current settings.

system session timeout custom rule rule-number protocol

Specifies the protocol to match for custom session timeout.

```
set system session timeout custom rule   rule-number protocol string
```

```
delete system session timeout custom rule   rule-number
```

```
show system session timeout custom rule  rule-number
```

rule-number

Number to identify the custom rule. The rules are evaluated in numeric order. The first match is used to apply the timeout value.

string

Protocol to match (alphanumeric string).

Configuration mode

```
system {
    session {
       timeout {
          custom {
            rule rule-number {
                 protocol string
             }
          }
       }
   }
}
```

Use this command to specify a protocol to match for custom session timeout.

Use the `set` form of this command to specify custom parameters.

Use the `delete` form of this command to remove custom settings.

Use the `show` form of this command to view the current settings.

system session timeout custom rule rule-number source

Specifies source parameters for custom session timeout.

```
set system session timeout custom rule  rule-number source { address address |
port port-number }
```

```
delete system session timeout custom rule  rule-number
```

```
show system session timeout custom rule  rule-number
```

rule-number

Number to identify the custom rule. The rules are evaluated in numeric order. The first match is used to apply the timeout value.

address

Destination IP address, subnet, or address group.

port

Destination port or port group.

Configuration mode

```
system {
    session {
       timeout {
          custom {
            rule rule-number {
               source {
                 address address
                 port port
                 }
             }
          }
       }
   }
}
```

Use this command to specify custom source parameters for session timeouts.

Use the `set` form of this command to specify custom parameters.

Use the `delete` form of this command to remove custom settings.

Use the `show` form of this command to view the current settings.

system session timeout icmp established

Sets the timeout for ICMP connections that are in the “established” state.

```
set system session timeout icmp established  timeout
```

```
delete system session timeout icmp established
```

```
show system session timeout icmp established
```

60 seconds

timeout

The amount of time, in seconds, that an ICMP connection waits in the “established” state before timing out. The timeout ranges from 1 through 21474836.

Configuration mode

```
system {
    session {
       timeout {
          icmp {
             established timeout {
             }
          }
       }
   }
}
```

Use the `set` form of this command to set the timeout for ICMP connections that are in the “established” state.

Use the `delete` form of this command to restore the default timeout.

Use the `show` form of this command to display the current timeout.

system session timeout icmp new

Sets the timeout for ICMP connections that are in the “new” state.

```
set system session timeout icmp new  timeout
```

```
delete system session timeout icmp new
```

```
show system session timeout icmp new
```

30 seconds

timeout

The amount of time, in seconds, that an ICMP connection waits in the “new” state before timing out. The timeout ranges from 1 through 21474836.

Configuration mode

```
system {
    session {
       timeout {
          icmp {
             new timeout {
             }
          }
       }
    }
}
```

Use the `set` form of this command to set the timeout for ICMP connections that are in the “new” state.

Use the `delete` form of this command to restore the default timeout.

Use the `show` form of this command to display the current timeout.

system session timeout other established

Defines the timeout for connections that use protocols other than ICMP, TCP, or UDP and are in the “established” state.

```
set system session timeout other established  timeout
```

```
delete system session timeout other established
```

```
show system session timeout other established
```

60 seconds

timeout

The amount of time, in seconds, that a connection waits in the “established” state before timing out. The timeout ranges from 1 to 21474836.

Configuration mode

```
system {
   session {
      timeout {
         other {
            established timeout {
            }
         }
      }
   }
}
```

Use the `set` form of this command to set the timeout for connections that use protocols other than ICMP, TCP, and UDP and are in the “established state.

Use the `delete` form of this command to restore the default timeout.

Use the `show` form of this command to display the current timeout.

system session timeout other new

Sets the timeout for connections that use protocols other than ICMP, TCP, and UDP and are in the “new” state.

```
set system session timeout other new  timeout
```

```
delete system session timeout other new
```

```
show system session timeout other new
```

30 seconds

timeout

The amount of time, in seconds, that a connection waits in the “new” state before timing out. The timeout ranges from 1 through 21474836.

Configuration mode

```
system {
    session {
       timeout {
          other {
             new timeout {
             }
          }
       }
    }
}
```

Use the `set` form of this command to set the timeout for connections that use protocols other than ICMP, TCP, or UDP and are in the “new” state.

Use the `delete` form of this command to restore the default timeout.

Use the `show` form of this command to display the current timeout.

system session timeout tcp close-wait

Sets the timeout for TCP connections that are in the "close-wait" state.

```
set system session timeout tcp close-wait  timeout
```

```
delete system session timeout tcp close-wait
```

```
show system session timeout tcp close-wait
```

21,600 seconds

timeout

The amount of time, in seconds, that a TCP connection waits in the “close-wait” state before timing out. The timeout ranges from 1 through 21474836.

Configuration mode

```
system {
    session {
       timeout {
          tcp {
             close-wait size {
             }
          }
       }
   }
}
```

Use the `set` form of this command to set the timeout for TCP connections that are in the “close-wait” state.

Use the `delete` form of this command to restore the default timeout.

Use the `show` form of this command to display the current timeout.

system session timeout tcp closed

Sets the timeout for TCP connections that are in the “closed” state.

```
set system session timeout tcp closed  timeout
```

```
delete system session timeout tcp closed
```

```
show system session timeout tcp closed
```

10 seconds

timeout

The amount of time, in seconds, a TCP connection waits in the “closed” state before timing out. The timeout ranges from 1 to 21474836.

Configuration mode

```
system {
    session {
       timeout {
          tcp {
             closed timeout {
             }
          }
       }
```

```
   }
}
```

Use the `set` form of this command to set the timeout for TCP connections that are in the “closed” state.

Use the `delete` form of this command to restore the default timeout.

Use the `show` form of this command to display the current timeout.

system session timeout tcp closing

Sets the timeout for TCP connections that are in the “closing” state.

```
set system session timeout tcp closing  timeout
```

```
delete system session timeout tcp closing
```

```
show system session timeout tcp closing
```

30 seconds

timeout

The amount of time, in seconds, a TCP connection waits in the “closing” state before timing out. The timeout ranges from 1 to 21474836.

Configuration mode

```

system {
   session {
      timeout {
         tcp {
            closing timeout {
            }
         }
      }
   }
}
```

Use the `set` form of this command to set the timeout for TCP connections that are in the “closing” state.

Use the `delete` form of this command to restore the default timeout.

Use the `show` form of this command to display the current timeout.

system session timeout tcp established

Sets the timeout for TCP connections that are in the “established” state.

```
set system session timeout tcp established  timeout
```

```
delete system session timeout tcp established
```

```
show system session timeout tcp established
```

86,400s

timeout

The amount of time, in seconds, a TCP connection waits in the “established” state before timing out. The timeout ranges from 1 to 21474836.

Configuration mode

```
system {
    session {
       timeout {
          tcp {
             established timeout {
             }
          }
       }
   }
}
```

Use the `set` form of this command to set the timeout for TCP connections that are in the “established” state.

Use the `delete` form of this command to restore the default timeout.

Use the `show` form of this command to display the current timeout.

system session timeout tcp fin-received

Sets the timeout for TCP connections that are in the “fin-received” state.

```
set system session timeout tcp fin-received  timeout
```

```
delete system session timeout tcp fin-received
```

```
show system session timeout tcp fin-received
```

240 seconds

timeout

The amount of time, in seconds, a TCP connection waits in the “fin-received” state before timing out. The timeout ranges from 1 to 21474836.

Configuration mode

```
system {
    session {
       timeout {
          tcp {
             fin-received timeout {
             }
          }
      }
   }
}
```

Use the `set` form of this command to set the timeout for TCP connections that are in the “fin-received” state.

Use the `delete` form of this command to restore the default timeout.

Use the `show` form of this command to display the current timeout.

system session timeout tcp fin-sent

Sets the timeout for TCP connections that are in the “fin-sent” state.

`set system session timeout tcp fin-sent` *`timeout`*

`delete system session timeout tcp fin-sent`

`show system session timeout tcp fin-sent`

240 seconds

timeout

The amount of time, in seconds, a TCP connection waits in the “fin-sent” state before timing out. The timeout ranges from 1 to 21474836.

Configuration mode

```
system {
   session {
      timeout {
         tcp {
            fin-sent timeout
         }
      }
   }
}
```

Use the `set` form of this command to set the timeout for TCP connections that are in the “fin-sent” state.

Use the `delete` form of this command to restore the default timeout.

Use the `show` form of this command to display the current timeout.

system session timeout tcp fin-wait

Sets the timeout for TCP connections that are in the “fin-wait” state.

```
set system session timeout tcp fin-wait  timeout
```

```
delete system session timeout tcp fin-wait
```

```
show system session timeout tcp fin-wait
```

21600 seconds

timeout

The amount of time, in seconds, a TCP connection waits in the “fin-wait” state before timing out. The timeout ranges from 1 to 21474836.

Configuration mode

```
system {
   session {
      timeout {
         tcp {
            fin-wait timeout
         }
      }
   }
}
```

Use the `set` form of this command to set the timeout for TCP connections that are in the “fin-wait” state.

Use the `delete` form of this command to restore the default timeout.

Use the `show` form of this command to display the current timeout.

system session timeout tcp last-ack

Sets the timeout for TCP connections that are in the “last-ack” state.

```
set system session timeout tcp last-ack  timeout
```

```
delete system session timeout tcp last-ack
```

```
show system session timeout tcp last-ack
```

30 seconds

timeout

The amount of time, in seconds, a TCP connection waits in the “last-ack” state before timing out. The timeout ranges from 1 to 21474836.

Configuration mode

```
system {
     session {
        timeout {
           tcp {
              last-ack timeout {
             }
          }
      }
   }
}
```

Use the `set` form of this command to set the timeout for TCP connections that are in the “last-ack” state.

Use the `delete` form of this command to restore the default timeout.

Use the `show` form of this command to display the current timeout.

system session timeout tcp simsyn-sent

Sets the timeout for TCP connections that are in the “simsyn-sent” state.

```
set system session timeout tcp simsyn-sent  timeout
```

```
delete system session timeout tcp simsyn-sent
```

```
show system session timeout tcp simsyn-sent
```

30 seconds

timeout

The amount of time, in seconds, a TCP connection waits in the “simsyn-sent” state before timing out. The timeout ranges from 1 to 21474836.

Configuration mode

```
system {
    session {
       timeout {
          tcp {
             simsyn-sent timeout {
             }
          }
       }
    }
}
```

Use the `set` form of this command to set the timeout for TCP connections that are in the "simsyn-sent" state.

Use the `delete` form of this command to restore the default timeout.

Use the `show` form of this command to display the current timeout.

system session timeout tcp syn-received

Sets the timeout for TCP connections that are in the "syn-received" state.

```
set system session timeout tcp syn-received  timeout
```

```
delete system session timeout tcp syn-received
```

```
show system session timeout tcp syn-received
```

60 seconds

timeout

The amount of time, in seconds, a TCP connection waits in the "syn-received" state before timing out. The timeout ranges from 1 to 21474836.

Configuration mode

```
system {
   session {
      timeout {
         tcp {
            syn-received timeout {
            }
         }
      }
   }
}
```

Use the `set` form of this command to set the timeout for TCP connections that are in the "syn-received" state.

Use the `delete` form of this command to restore the default timeout.

Use the `show` form of this command to display the current timeout.

system session timeout tcp syn-sent

Sets the timeout for TCP connections that are in the "syn-sent" state.

```
set system session timeout tcp syn-sent  timeout
```

```
delete system session timeout tcp syn-sent
```

```
show system session timeout tcp syn-sent
```

30 seconds

timeout

The amount of time, in seconds, a TCP connection waits in the “syn-sent” state before timing out. The timeout ranges from 1 to 21474836.

Configuration mode

```
system {
   session {
      timeout {
         tcp {
            syn-sent timeout {
            }
         }
      }
   }
}
```

Use the `set` form of this command to set the timeout for TCP connections that are in the “syn-sent” state.

Use the `delete` form of this command to restore the default timeout.

Use the `show` form of this command to display the current timeout.

system session timeout tcp time-wait

Sets the timeout for TCP connections that are in the “time-wait” state.

```
set system session timeout tcp time-wait  timeout
```

```
delete system session timeout tcp time-wait
```

```
show system session timeout tcp time-wait
```

21600 seconds

timeout

The amount of time, in seconds, a TCP connection waits in the “time-wait” state before timing out. The timeout ranges from 1 to 21474836.

Configuration mode

```
system {
    session {
       timeout {
          tcp {
             time-wait timeout {
             }
          }
       }
    }
}
```

Use the `set` form of this command to set the timeout for TCP connections that are in the “time-wait” state.

Use the `delete` form of this command to restore the default timeout.

Use the `show` form of this command to display the current timeout.

system session timeout udp established

Sets the timeout for UDP connections that are in the “established” state.

```
set system session timeout udp established  timeout
```

```
delete system session timeout udp established
```

```
show system session timeout udp established
```

60 seconds

timeout

The amount of time, in seconds, a UDP connection waits in the “established” state before timing out. The timeout ranges from 1 to 21474836.

Configuration mode

```
system {
    session {
       timeout {
          udp {
             established timeout {
             }
          }
       }
   }
}
```

Use the `set` form of this command to set the timeout for UDP connections that are in the “established” state.

Use the `delete` form of this command to restore the default timeout.

Use the `show` form of this command to display the current timeout.

system session timeout udp new

Sets the timeout for UDP connections that are in the “new” state.

```
set system session timeout udp new  timeout
```

```
delete system session timeout udp new
```

```
show system session timeout udp new
```

30 seconds

timeout

The amount of time, in seconds, a UDP connection waits in the “new” state before timing out. The timeout ranges from 1 to 21474836.

Configuration mode

```
system {
   session {
      timeout {
         udp {
            new timeout {
            }
         }
      }
   }
}
```

Use the `set` form of this command to set the timeout for UDP connections that are in the “new” state.

Use the `delete` form of this command to restore the default timeout.

Use the `show` form of this command to display the current timeout.

show session table detail

Shows the dataplane session table details.

```
show session table  detail [ source src-addr [ : src-port] ]
```

src-addr [: src-port]

Source IP address with optional port. Filters the output based on either a source IP address or a source IP address combined with a port.

Operational mode

Use this command to view details of the dataplane session table.

show session table application

Shows the dataplane session table application information.

```
show session table  application [ source src-addr [ : src-port] ]
```

src-addr [: src-port]

Source IP address with optional port. Filters the output based on either a source IP address or a source IP address combined with a port.

Operational mode

Use this command to view application information of the dataplane session table.

system static-host-mapping host-name

Statically maps to a host name and an IP address and one or more aliases.

```
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 ]
```

name

Multi-node. A Fully Qualified Domain Name (FQDN) name being statically mapped to an IP address; for example, router1@mydomain.com. The name can contain only letters, numbers, periods (.), and hyphens (-).

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. The name can contain only letters, numbers, and hyphens (-).

You can define multiple aliases for a host name by creating multiple alias configuration nodes.

Configuration mode

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

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 map 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 entry 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

Sets the time zone for the local system clock.

```
set system time-zone  zone
```

```
delete system time-zone
```

```
show system time-zone
```

The default time zone is Greenwich mean time (GMT).

zone

A time zone in the format of *region/location*; for example, **US/Pacific**Note that both *region* and *location* are case sensitive. Use command completion (that is, the <Tab> key) to display available time zones.

Configuration mode

```
system {
    time-zone zone
}
```

Use this command to set the time zone for the local system clock. To set the time, you specify a region and location. Use command completion (that is, the <Tab> key) to display time zones that are available.

In addition to the wide range of time zones available, backward compatibility is achieved by using Etc/<offset> and SystemV/<offset> as *region/location*. Note that Etc/<offset> uses Posix-style offsets. These offsets use plus 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 command restores the time zone to the default (GMT).

Use the `show` form of this command to view the time zone.

Chapter 10. Role-based Access Control

This chapter explains role-based access control (RBAC) and how to configure this feature.

Overview

Role-based Access Control (RBAC) is a method of restricting access to part of the configuration to authorized users. RBAC allows an administrator to define the rules for a group of users that restrict which commands users of that group are allowed to run.

RBAC is performed by first creating a group assigned to the Access Control Management (ACM) rule set, adding a user to the group, creating a rule set to match the group to the paths in the system, then configuring the system to allow or deny those paths that are applied to the group.

Users are allowed to be in one of three class of users with defined privilege levels:

- *Operator*—Allowed to execute commands that are defined in the Vyatta CLI. Not allowed to into config mode.
- *Administrator*—Allowed to execute arbitrary Linux commands in addition to commands that are defined by the Vyatta CLI and to enter configuration mode.
- *Superuser*—Allowed to execute commands with root privileges through the `sudo` command in addition to having administrator class privileges.

By default, all users that are defined to be in the superuser or the administrator class belong to a common group called vyattacfg. This group allows a rule set to be defined that pertains to both the superuser and administrator classes without defining two group matches. The operator class users belong to the vyattaop group.

router allows a superuser to create new groups based on your requirements. IP Infusion Inc. recommends creating a group with the highest level of privileges, called a *security* group. A superuser can set rules so that only members of the *security* group are allowed to modify the ACM and login information. This prevents administrators from inadvertently compromising the system image or the ACM list.

Path matching

System configuration is modeled after a tree structure and enables the user to filter any path of that tree. The system supports only absolute addressing that begins with / as the root and uses the wildcard operator (*) as the path language.

Operational mode paths are absolute and do not match their children if a wildcard operator (*) is not included at the end of the path. Therefore, not using the wildcard operator restricts the user to specific commands.

In the following example, rule 1 restricts the use of the `show` command to only `show interfaces` and rule 2 denies all other `show` commands.

```
rule 1 {
    action allow
    path "/show/interfaces"
}
rule 2 {
    action deny
    path "/show/*"
}
```

Default rule set

The router is preconfigured with a default rule set for RBAC. The following example shows the default rule set in RBAC.

```
super@vyatta# show system acm
 create-default deny
 delete-default deny
 enable
 exec-default allow
 operational-ruleset {
     rule 9988 {
         action deny
         command /show/configuration
         group vyattaop
     }
     rule 9989 {
         action allow
         command "/clear/*"
         group vyattaop
     }
     rule 9990 {
         action allow
         command "/show/*"
         group vyattaop
     }
     rule 9991 {
         action allow
         command "/monitor/*"
         group vyattaop
     }
     rule 9992 {
         action allow
         command "/ping/*"
         group vyattaop
     }
     rule 9993 {
         action allow
         command "/reset/*"
         group vyattaop
     }
     rule 9994 {
         action allow
         command "/release/*"
         group vyattaop
     }
     rule 9995 {
         action allow
         command "/renew/*"
         group vyattaop
     }
     rule 9996 {
         action allow
         command "/telnet/*"
         group vyattaop
     }
     rule 9997 {
         action allow
         command "/traceroute/*"
         group vyattaop
     }
     rule 9998 {
         action allow
         command "/update/*"
         group vyatta-op
     }
     rule 9999 {
         action deny
         command "*"
         group vyattaop
     }
 }
 read-default allow
 ruleset {
     rule 9999 {
         action allow
         group vyattacfg
         operation "*"
         path "*"
     }
 }
 update-default deny
}
```

Configuration examples

As an example of RBAC configuration, this section shows how to add to the default rule set and create a new role for users who should be allowed to access information regarding only routing protocols on the system. Essentially, rules are being defined for a group of users that restrict which commands the users of that group are allowed to run.

Example of a rule set in operational mode

Operational mode has a rule set like the configuration mode that allows administrators to specify which operation mode commands a user is allowed to run. For example, as a protocol administrator, the user needs to execute only the `show interfaces` and `show ip` families of commands and, therefore, should not be allowed to run other administrative actions.

To define the operation mode rules for the protocol administrator group (protoadmin), perform the following steps in configuration mode.

Table 28. Defining the operational mode rules for the protocol administrator group

Step	Description	Command

1	Create a rule allowing all operations on /show/ip for the protoadmin group.	`vyatta@R1#set system acm operational-ruleset rule 10 action 'allow'`
		`vyatta@R1#set system acm operational-ruleset rule 10 command '/show/ip/*'`
		`vyatta@R1#set system acm operational-ruleset rule 10 group 'protoadmin'`

2	Create a rule allowing all operations on /show/interfaces for the protoadmin group.	`vyatta@R1#set system acm operational-ruleset rule 20 action 'allow'`
		`vyatta@R1#set system acm operational-ruleset rule 20 command '/show/interfaces/*'`
		`vyatta@R1#set system acm operational-ruleset rule 20 group 'protoadmin'`

3	Create a rule allowing all operations on /configure for the protoadmin group.	`vyatta@R1#set system acm operational-ruleset rule 30 action 'allow'`
		`vyatta@R1#set system acm operational-ruleset rule 30 command '/configure'`
		`vyatta@R1#set system acm operational-ruleset rule 30 group 'protoadmin'`

4	Deny all operations on all other paths for the protoadmin group.	`vyatta@R1#set system acm operational-ruleset rule 40 action 'deny'`
		`vyatta@R1#set system acm operational-ruleset rule 40 command '*'`
		`vyatta@R1#set system acm operational-ruleset rule 40 group 'protoadmin'`

The following example shows the operational mode rule set that is configured in Table 28: Defining the operational mode rules for the protocol administrator group.

```
super@vyatta# show system acm operational-ruleset
rule 10 {
    action allow
    command "/show/ip/*"
    group protoadmin
}
rule 20 {
    action allow
    command "/show/interfaces/*"
    group protoadmin
}
rule 30 {
    action allow
    command /configure
    group protoadmin
}
rule 40 {
    action deny
    command "*"
    group protoadmin
}
```

The following example shows system login information regarding the protoadmin group with a user called john as a member of that group.

```

super@vyatta# show system login
group protoadmin {
}
user john {
authentication {
encrypted-password *******
}
group protoadmin
level admin
}
super@vyatta#
```

Rule set in operation

After logging in as a user, the configuration command options are filtered to allow only what the user can access based on the permissions for the user. Output from the `show` command for the user configuration is also filtered.

This section displays the filtered output for a user called john in the protoadmin group. Notice that this user is restricted to the interfaces, policy, and protocols configuration commands as configured in the following example.

```
john@vyatta# set <tab>
 Possible completions:
 > interfaces   Network interfaces
 > policy       PBR, QoS, & routing policy
 > protocols    Routing protocol parameters
```

In the following example, the resources, security, service, and system branches of the tree are missing, which indicates that the configuration command options for these branches are not available to the user called john.

```

[edit]
john@vyatta# show
 interfaces {
     dataplane dp0p2p1 {
         address dhcp
         description "foo bar"
         mtu 1500
     }
     dataplane dp0port2 {
         address dhcp
         mtu 1500
     }
     loopback lo {
     }
 }
policy {
    route {
        route-map test {
            rule 10 {
            action permit
            }
        }
    }
}
protocols {
     static {
         route 198.18.1.2/32 {
             next-hop 198.18.2.3 {
             }
         }
     }
 }
[edit]
john@vyatta#
```

Example of a rule set in configuration mode

To manage the routing protocols on the system, the user needs access to only the interface and the routing protocol subtrees in the configuration.

To configure RBAC, you must add the protocol administrator role or group.

To add the protocol administrator group and define the rules for this group of users, perform the following steps in configuration mode.

Table 29. Adding a protocol administrator group and defining the rules for the group

Step	Description	Command
1	Create a protocol administrator group.	vyatta@R1# `set system login group protoadmin`
2	Add a user to the group.	vyatta@R1# `set system login user johngroup protoadmin`

3	Create a rule that allows all operations on /protocols.	`vyatta@R1#set system acm ruleset rule 10 action allow`
		`vyatta@R1#set system acm ruleset rule 10 group protoadmin`
		`vyatta@R1#set system acm ruleset rule 10 operation *`
		`vyatta@R1#set system acm ruleset rule 10 path /protocols`

4	Create a rule that allows all operations on /policy.	`vyatta@R1#set system acm ruleset rule 20 action allow`

Table 29. Adding a protocol administrator group and defining the rules for the group (continued)

Step	Description	Command
		vyatta@R1#set system acm ruleset rule 20 group protoadmin vyatta@R1#set system acm ruleset rule 20 operation * vyatta@R1#set system acm ruleset rule 20 path /policy
5	Create a rule that allows all operations on /interfaces.	vyatta@R1#set system acm ruleset rule 30 action allow vyatta@R1#set system acm ruleset rule 30 group protoadmin vyatta@R1#set system acm ruleset rule 30 operation * vyatta@R1#set system acm ruleset rule 30 path /interfaces
6	Deny all operations on all other paths for users of the protoadmin group.	vyatta@R1#set system acm ruleset rule 40 action deny vyatta@R1#set system acm ruleset rule 40 group protoadmin vyatta@R1#set system acm ruleset rule 40 operation * vyatta@R1#set system acm ruleset rule 40 path *

The following example shows the configuration mode rule set that is configured in Table 29: Adding a protocol administrator group and defining the rules for the group.

```
super@vyatta# show system acm ruleset
rule 10 {
    action allow
    group protoadmin
    operation "*"
    path /protocols
}
rule 20 {
    action allow
    group protoadmin
    operation "*"
    path /policy
}
rule 30 {
    action allow
    group protoadmin
    operation "*"
    path /interfaces
}
rule 40 {
    action deny
    group protoadmin
    operation "*"
    path "*"
}
```

The following example shows system login information regarding the protoadmin group with a user called john as a member of that group.

```
super@vyatta# show system login
group protoadmin {
}
user john {
authentication {
```

```
encrypted-password *******
}
group protoadmin
level admin
}
super@vyatta#
```

Rule set in operation

After logging in as a user, the operational mode command options are filtered to allow only what the user can access based on the permissions for the user.

The following example displays the filtered output for a user called john in the protoadmin group. This example shows a subset of operational mode paths to which this user has been given access.

```
john@vyatta$ <tab>
Possible completions:
  configure     Enter configure mode
  show          Show system information
john@vyatta$
```

The following example shows that the user called john is limited to the specific `show` commands with access to only the `show interfaces` and `show ip` families of commands.

```
john@vyatta# run show <tab>
Possible completions:
  interfaces    Show network interface information
  ip            Show IPv4 routing informationjohn@vyatta$ show <tab>
```

Example of a rule set to create a security group

Consider a router where a superuser creates a new group called *security*. The superuser associates a rule set with the new group so that only members of this group can modify the ACM and login information. Additionally, a member called *secadmin*, who is part of the administrator group, is allowed to be a part of this new group.

To create the new group and to associate the rule set, perform the following steps in configuration mode.

Step	Command
Create a group called *security*. Members of the group are allowed to adjust the security policy and system logins.	`vyatta@vyatta# set system login group 'security'`

Promote a member called *secadmin* from the administrator group to the security group.	`vyatta@vyatta# set system login user secadmin`
	`authentication plaintext-password #<enter>; enter`
	`password`
	`vyatta@vyatta# set system login user secadmin group`
	`'security'`

Allow the members of the security group access to all possible operations.	

Step	Command

	`vyatta@vyatta# set system acm ruleset rule 1 action 'allow'`
	`vyatta@vyatta# set system acm ruleset rule 1 group 'security'`
	`vyatta@vyatta# set system acm ruleset rule 1 operation '*'`
	`vyatta@vyatta# set system acm ruleset rule 1 path '*'`

Prohibit changes to `/system/acm and /system/login` unless the changes are made by a member of the group called *security*.	`vyatta@vyatta# set system acm ruleset rule 9991 group 'vyattacfg'`
	`vyatta@vyatta# set system acm ruleset rule 9991 operation 'delete'`
	`vyatta@vyatta# set system acm ruleset rule 9991 path '/system/acm'`
	
	`vyatta@vyatta# set system acm ruleset rule 9992 group 'vyattacfg'`
	`vyatta@vyatta# set system acm ruleset rule 9992 operation 'create'`
	`vyatta@vyatta# set system acm ruleset rule 9992 path '/system/acm'`
	
	`vyatta@vyatta# set system acm ruleset rule 9993 group 'vyattacfg'`
	`vyatta@vyatta# set system acm ruleset rule 9993 operation 'update'`
	`vyatta@vyatta# set system acm ruleset rule 9993 path '/system/acm'`
	
	`vyatta@vyatta# set system acm ruleset rule 9994 group 'vyattacfg'`
	`vyatta@vyatta# set system acm ruleset rule 9994 operation 'update'`
	`vyatta@vyatta# set system acm ruleset rule 9994 path '/system/login'`
	
	`vyatta@vyatta# set system acm ruleset rule 9995 group 'vyattacfg'`
	`vyatta@vyatta# set system acm ruleset rule 9995 operation 'delete'`
	`vyatta@vyatta# set system acm ruleset rule 9995 path '/system/login'`
	
	`vyatta@vyatta# set system acm ruleset rule 9996 group 'vyattacfg'`
	`vyatta@vyatta# set system acm ruleset rule 9996 operation 'create'`
	`vyatta@vyatta# set system acm ruleset rule 9996 path '/system/login'`

The following rule set is displayed by entering the `show acm` command in operational mode after you perform the steps in the preceding section.

```
# show system acm
 acm {
     enable
     operational-ruleset {
         rule 9977 {
             action allow
             command /show/tech-support/save
             group vyattaop
         }
         rule 9978 {
             action deny
             command "/show/tech-support/save/*"
             group vyattaop
         }
         rule 9979 {
             action allow
             command /show/tech-support/save-uncompressed
             group vyattaop
```

```
        }
        rule 9980 {
            action deny
            command "/show/tech-support/save-uncompressed/*"
            group vyattaop
        }
        rule 9981 {
            action allow
            command /show/tech-support/brief/save
            group vyattaop
        }
        rule 9982 {
            action deny
            command "/show/tech-support/brief/save/*"
            group vyattaop
        }
        rule 9983 {
            action allow
            command /show/tech-support/brief/save-uncompressed
            group vyattaop
        }
        rule 9984 {
            action deny
            command "/show/tech-support/brief/save-uncompressed/*"
            group vyattaop
        }
        rule 9985 {
            action allow
            command /show/tech-support/brief/
            group vyattaop
        }
        rule 9986 {
            action deny
            command /show/tech-support/brief
            group vyattaop
        }
        rule 9987 {
            action deny
            command /show/tech-support
            group vyattaop
        }
        rule 9988 {
            action deny
            command /show/configuration
            group vyattaop
        }
        rule 9989 {
            action allow
            command "/clear/*"
            group vyattaop
        }
        rule 9990 {
            action allow
```

```
                command "/show/*"
                group vyattaop
            }
            rule 9991 {
                action allow
                command "/monitor/*"
                group vyattaop
            }
            rule 9992 {
                action allow
                command "/ping/*"
                group vyattaop
            }
            rule 9993 {
                action allow
                command "/reset/*"
                group vyattaop
            }
            rule 9994 {
                action allow
                command "/release/*"
                group vyattaop
            }
            rule 9995 {
                action allow
                command "/renew/*"
                group vyattaop
            }
            rule 9996 {
                action allow
                command "/telnet/*"
                group vyattaop
            }
            rule 9997 {
                action allow
                command "/traceroute/*"
                group vyattaop
            }
            rule 9998 {
                action allow
                command "/update/*"
                group vyattaop
            }
            rule 9999 {
                action deny
                command "*"
                group vyattaop
            }
        }
        ruleset {
            rule 1 {
                action allow
                group security
                operation "*"
                path "*"
            }
            rule 9991 {
                group vyattacfg
                operation delete
                path /system/acm
            }
            rule 9992 {
                group vyattacfg
                operation create
                path /system/acm
            }
            rule 9993 {
                group vyattacfg
                operation update
                path /system/acm
            }
            rule 9994 {
                group vyattacfg
                operation update
                path /system/login
            }
            rule 9995 {
                group vyattacfg
                operation delete
                path /system/login
            }
            rule 9996 {
                group vyattacfg
                operation create
                path /system/login
            }
            rule 9999 {
                action allow
                group vyattacfg
                operation "*"
                path "*"
            }
        }
    }
```

Rule set in operation for the security group

After the security group is created, non members of the group are unable to change the ACM or login information, even if they are members of the administrator group.

Consider two users, *secadmin* and *cosadmin*, who belong to the administrator group. *Secadmin* is a member of the security group. *Cosadmin* is not a member of the security group.

As a member of the security group, *secadmin* can promote himself to a superuser. The following is an example of the login of a user called *secadmin* who is a member of the *security* group:

```
secadmin@vyatta:~$ configure
secadmin@vyatta# set system login user secadmin level superuser
secadmin@vyatta# commit
```

The following is an example of the login of a user called *cosadmin* who is not a member of the *security* group.

```
cosadmin@vyatta# set system login user cosadmin level superuser
access denied
```

Chapter 11. Role-based Access Control Commands

system acm create-default

Specifies the default action for the create operation.

```
set system acm create-default { allow | deny }
```

```
delete system acm create-default { allow | deny }
```

```
show system acm create-default
```

By default, the create operation is denied.

allow

Allows the operation.

deny

Denies the operation.

Configuration mode

```
acm {
    create-default {
        allow
        deny
     }
}
```

Use the `set` form of this command to specify the default action for the create operation.

Use the `delete` form of this command to delete the specified default action for the create operation.

Use the `show` form of this command to display the specified default action for the create operation.

system acm delete-default

Specifies the default action for the delete operation.

```
set system acm delete-default { allow | deny }
```

```
delete system acm delete-default { allow | deny }
```

```
show system acm delete-default
```

By default, the delete operation is denied.

allow

Allows the operation.

deny

Denies the operation.

Configuration mode

```
system {
    delete-default {
        allow
        deny
      }
}
```

Use the `set` form of this command to specify the default action for the delete operation.

Use the `delete` form of this command to delete the specified default action for the delete operation.

Use the `show` form of this command to display the specified default action for the delete operation.

system acm enable

Enables the ACM rule sets.

```
set system acm enable
```

```
delete system acm enable
```

```
show system acm enable
```

Configuration mode

```
system {
     acm {
        enable
     }
}
```

Use the `set` form of this command to enable the ACM rule sets.

Use the `delete` form of this command to disable the ACM rule sets.

Use the `show` form of this command to display the ACM rule sets.

system acm exec-default

Specifies the default action for the execute operation.

```
set system acm exec-default { allow | deny }
```

```
delete system acm exec-default { allow | deny }
```

```
show system acm exec-default
```

By default, the execute operation is allowed.

allow

Allows the operation.

deny

Denies the operation.

Configuration mode

```
system {
    acm {
        exec-default
         allow
         deny
     }
}
```

Use the `set` form of this command to specify the default action for the execute operation.

Use the `delete` form of this command remove the default action for the execute operation.

Use the `show` form of this command display default action for the execute operation.

system acm operational-ruleset rule

Enables an operational command rule set for ACM.

```
set system acm operational-ruleset rule  [ number ]
```

```
delete system acm operational-ruleset rule [ number ]
```

```
show system acm operational-ruleset rule  [ number ]
```

number

A rule number. The number ranges from 1 through 9999.

Configuration mode

```
system {
    acm {
        operational-ruleset {
                 rule number
        }
     }
}
```

Use the `set` form of this command to enable an operational rule for ACM.

Use the `delete` form of this command to disable an operational rule for ACM.

Use the `show` form of this command to display an operational rule for ACM.

system acm read-default

Specifies the default action for the read operation.

```
set system acm read-default  { allow | deny }
```

```
delete system acm read-default { allow | deny }
```

```
show system acm read-default
```

By default, the read operation is allowed.

allow

Allows the operation.

deny

Denies the operation.

Configuration mode

```

system {
   acm {
       read-default {
          allow
          deny
       }
    }
}
```

Use the `set` form of this command to specify the default action for the read operation.

Use the `delete` form of this command to disable the specified default action for the read operation.

Use the `show` form of this command to display the specified default action for the read operation.

system acm ruleset rule action

Specifies the action to be taken for a specified ACM rule set.

```
set system acm ruleset rule  number action { allow | deny }
```

```
delete system acm ruleset rule  number action { allow | deny }
```

```
show system acm ruleset rule  number action
```

allow

Allows the operation.

deny

Denies the operation.

Configuration mode

```
system {
    acm {
        ruleset {
            rule number {
                action {
                    allow
                    deny
                }
            }
        }
    }
}
```

Use the `set` form of this command to specify the action to be taken for a specified rule set.

Use the `delete` form of this command to delete the specified action for an ACM rule set.

Use the `show` form of this command to display the actions settings for an ACM rule set.

system acm ruleset rule group

Defines a group operation to match for an ACM rule.

```
set system acm ruleset rule  number group group-name
```

```
set system acm ruleset rule  number group group-name
```

```
set system acm ruleset rule  number group group-name
```

number

A rule number. The number range from 1 through 9999.

group-name

A group to match.

Configuration mode

```
system {
    acm {
        rule number
            group group-name
    }
}
```

Use the `set` form of this command to define a group operation to match for a ACM rule.

Use the `delete` form of this command to remove a group operation to match.

Use the `show` form of this command to display a group operation to match.

system acm ruleset rule log

Defines the log operation on a ACM rule.

`set system acm ruleset rule` *`number`* **`log`**

`delete system acm ruleset rule` *`number`* **`log`**

`show system acm ruleset rule` *`number`* **`log`**

number

A rule set number. The number ranges from 1 through 9999.

Configuration mode

```
ruleset {
    rule number {
        log
     }
}
```

Use this command to define the log operation on a ACM rule.

system acm ruleset rule operation

Defines a path operation to match for an ACM rule.

`set system acm ruleset rule` *`number`* **`operation`** `{` **`create`** **`|`** **`delete`** **`|`** **`read`** **`|`** **`update`** **`|`** **`*`** `}`

`delete system acm ruleset rule` *number* **operation** [**create** | **delete** | **read** | **update** | *****]

`show system acm ruleset rule` *number*

number

A rule number. The number ranges from 1 through 9999.

create

Specifies a create path operation to match.

read

Specifies a read path operation to match.

update

Specifies an update path operation to match.

delete

Specifies a delete path operation to match.

*****

Specifies all paths operations to match.

Configuration mode

```
system {
    acm {
        ruleset {
                 rule number {
                    operation create
                    operation read
                    operation update
                    operation delete
                    operation *
             }
        }
   }
}
```

You must have the path configured for the rule to commit this configuration.

Use the `set` form of this command to define a path operation to match for an ACM rule.

Use the `delete` form of this command to remove the path operation to match.

Use the `show` form of this command to display the path operation to match.

system acm ruleset rule path

Defines a path to match for an ACM rule.

`set system acm ruleset rule` *number* **path** *path* `delete system acm ruleset rule` *`number`* **`path`** `[` *`path`* `]`

`show system acm ruleset rule` *`number`* **`path`**

number

A rule set number. The number ranges from 1 through 9999.

path

A path to match; for example, /protocols.

Configuration mode

```
system {
     acm {
          ruleset {
               rule number
          }
          path path
     }
}
```

Use the `set` form of this command to define a path to match for an ACM rule.

Use the `delete` form of this command to remove a path for an ACM rule.

Use the `show` form of this command to display the path for an ACM rule.

system acm update-default

Specifies the default action for the update operation.

`set system acm update-default` `{` **`allow`** `|` **`deny`** `}`

`delete system acm update-default` `{` **`allow`** `|` **`deny`** `}`

`show system acm update-default` `{` **`allow`** `|` **`deny`** `}`

By default, the update operation is denied.

allow

Allows the operation.

deny

Denies the operation.

Configuration mode

```
system {
    acm {
        update-default allow
        update-default deny
    }
}
```

Use the `set` form of this command to specify the default action for the update operation.

Use the `delete` form of this command to delete the specified default action for the update operation.

Use the `show` form of this command to display the specified default action for the update operation.

Chapter 12. User Management

This chapter explains how to set up user accounts and user authentication.

User management configuration

This section presents the following topics:

- User management overview
- Creating a login user account
- GRUB menu configuration options
- Configuring a system for a RADIUS authentication server
- Configuring a system for a TACACS+ authentication server
- Configuring a system for SSH access using shared public keys

User management overview

This section presents the following topics:

- Login authentication
- RADIUS authentication
- TACACS+ authentication
- SSH access using shared public keys

The router supports all 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 Controller 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: the `vyatta` user with the `vyatta password` . 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 established by using system login user authentication.

You can change user account information by 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 router the Linux `passwd` command can be used only 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.

A login password is supplied in plain text. After configuration is committed, the system encrypts the password 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 of 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 times out and must be repeated.

RADIUS authentication

A RADIUS server is used only to authenticate user passwords. Using RADIUS authentication does not affect the privilege level of a user. RADIUS authentication is not supported for IPv6.

To configure RADIUS, you specify the location of a RADIUS server and specify the secret to be used to authenticate the user on the RADIUS server. A RADIUS secret is specified in plain text. It is 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 a RADIUS secret, it is displayed in plain text. A RADIUS secret must not contain spaces and is 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, a user must still be configured in the Vyatta login database; otherwise, the user is not able to access the router and, therefore, is not able to query the RADIUS server.

TACACS+ authentication

This section presents the following topics:

- Mapping router user IDs to TACACS+ usernames
- Specifying authentication level in TACACS+
- Restricting access through connection type
- Troubleshooting TACACS+ authentication issues

TACACS+ is a distributed access control system for routers that provides authentication, authorization, and accounting.

To configure TACACS+, you specify the location of the TACACS+ server and specify the secret to be used to authenticate the user on the server. A TACACS+ secret is specified in plain text and stored in plain text on the system and is used as part of a cryptographic operation for transferring authentication information securely over the network. A TACACS+ secret must not contain spaces and is 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 login database of the router. A TACACS+ server can be used either as the only authentication server or as a supplement to the router, providing password authentication.

Mapping router user IDs to TACACS+ usernames

You can map a router local user ID to a different username recorded on a TACACS+ server. The mapping is specified on the TACACS+ server.

For example, to map to the `tac-user` username on the TACACS+ server to the `vyatta-user` username on the local router, the (partial) configuration on the TACACS+ server looks 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 router by using the `tac-user` account ID actually logs the user in to the router as `vyatta-user`.

Order of authentication

If the system is configured for authentication chaining, the order of authentication is based on the authentication chaining. For more information about the authentication chaining method, see Authentication chaining method.

If the system is not configured using the authentication chaining method, then 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 by 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 router authentication database. If local authentication fails, the access attempt is rejected.

Note: The login process itself has a 60-second timeout. If a user cannot be authenticated in this time by a configured authentication server, then the login attempt times out.

When the system is configured for TACACS+ and a user is configured on it and on the local user database, the login attempt fails if the user fails authentication on TACACS+. The local user database is used only when the user does not exist on the TACACS+ server or that server becomes unavailable.

Specifying authentication level in TACACS+

By default, TACACS+ authorized users on the router are given operator-level access. However, you can specify the authentication level for individual TACACS+ authorized users on the local router. Like the mapping of user IDs, this configuration is specified on the TACACS+ server, as shown in the following example:

```
user = administrator {
     default service = permit
     login = cleartext "vyatta"
     service = vyatta-exec {
        level = "admin"
}
}
```

Logging in to the local router as the `administrator` user in this instance provides administrative-level access. You can also configure an additional level on the TACACS+ server as `superuser` to provide superuser-level access.

Restricting access through connection type

The router sends different connection-type information through the TACACS+ protocol based on the type of connection by which the user is accessing the router. This information can be used to restrict how certain types of users are allowed to access the system. For example, it is possible to restrict administrators to only login access through the physical console rather than remotely through SSH or Telnet.

Table 30. Protocol values sent to TACACS+ based on connection type

Connection type	Protocol value sent to TACACS+
Console	login
SSH	sshd
Telnet	telnet

Troubleshooting TACACS+ authentication issues

Because TACACS+ requires a secret, data is encrypted and, therefore, debugging 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 by using `tshark` , given a secret of **mysecret** on the well-known TACACS+ port (**tacacs** , which is port 49), you enter either of the following commands:

```
tshark -o tacplus.key:mysecret tcp port tacacs

tshark -o tacplus.key:mysecret tcp port 49
```

SSH access using shared public keys

Remote access to the router is typically accomplished through Telnet or SSH. For either of these methods, passwords are authenticated by using the local login user database, a RADIUS server, or a TACACS+ server, as previously described. SSH is typically used when a secure session is required. One potential problem with password authentication, even by 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 by using shared public keys. With this authentication method, a private and public key pair are generated (typically by using the Linux `ssh-keygen` command) on a remote system. The public key file (typically with a extension) is loaded into the login configuration for the user who is accessing the system with it by using loadkey. In addition, the router must be configured to disable password authentication for SSH (refer to *Services Configuration Guide*). So, SSH users can be authenticated by using passwords or shared public keys, but not both.

Maintenance of SSH public keys of known hosts

The router uses the SSH client in various subsystems to allow secure data exchange or file transfer with other trusted systems in the network. The identity of SSH servers can be verified by an SSH public-key which gets checked upon each connection attempt by the SSH client. To prevent Man-in-the-Middle attacks, when a malicious system tries to act as the designated SSH server, the SSH public-key of the server gets verified on each connection attempt by the router.

How it works

The router uses a global known hosts database to maintain the public keys of trusted and known SSH hosts. This SSH known hosts database needs to be pre-populated with the trusted SSH public keys of the systems that the router is likely to interact with by means of SSH. The router administrator populates the database. On a connection attempt, if the SSH server public key of a known or trusted host is a mismatch, the router prevents any file or data exchange with the potentially malicious SSH server.

SSH known-hosts configuration on router subsystems

The following subsystems or functionality of the router rely on non-interactive SSH authorization where the SSH known-hosts of the target system need to be known:

- "system config-management commit-archive location" if configured for a "scp://" target
- Usage of the `copy` operational command with an "scp://" target
- All calls of SSH tools by router operators or administrators on the Vyatta shell

Maintenance of SSH known hosts database

The following configuration parameters are used to populate the global SSH known hosts database:

- `security ssh-known-hosts host` [*hostname*] `load-from-file` [*file*]
- `security ssh-known-hosts host` [*hostname*] **key** "[*key type*] [*base64 encoded key*]"
- `security ssh-known-hosts host` [*hostname*] `fetch-from-server`

Configuration example: public key loaded from a local file

A public key can be loaded from a local file using:

`security ssh-known-hosts host` [*hostname*] `load-from-file` [*file*]

where the `file` is a plain-text file holding the SSH public-key as generated by the `ssh-keyscan` command. (<server address> <key type> <base64 encoded key>)

```
vyatta@vyatta# set security ssh-known-hosts host 192.168.122.1
 load-from-file ~/192.168.122.1.pub
  Adding key for 192.168.122.1 with fingerprint:
  2048 60:9e:25:55:31:ee:c9:e9:73:a2:22:a8:18:b0:80:0e 192.168.122.1 (RSA)
```

Public key as base64 encoded key

If the key is available as a base64 encoded string, it can be also be imported to the database with the following configuration security parameter:

`ssh-known-hosts host` [*hostname*] **key** [*key type*] [*base64 encoded key*]

Note: The key type and the base64 encoded key need to be one quoted string.

```
vyatta@vyatta# set security ssh-known-hosts host 192.168.122.1 key “ssh-rsa
 AAAAB3NzaC1y...”
```

Import of SSH current public key from the server

The SSH current public key can be directly imported from the server by means of the network. This method fetches the SSH public key of the server from the target server on the given network. The SSH public key fetch is only done once initially. The SSH public key then gets stored persistently in the SSH known hosts database.

Note: We recommend that you use direct import only in a trusted network. This is to guarantee that on the initial fetch, no malicious system on the same network or in between performs a Man-in-the-middle attack.

```
vyatta@vyatta# set security ssh-known-hosts host 192.168.122.1
 fetch-from-server

  Adding key for 192.168.122.1 with fingerprint:
  2048 60:9e:25:55:31:ee:c9:e9:73:a2:22:a8:18:b0:80:0e 192.168.122.1 (RSA)
```

Creating a login user account

This section presents a sample configuration for a user account that is validated by using the local user database. Figure 6: Login user account shows the sample configuration.

Figure 6. Login user account

Table 31: Creating a login user account shows how to create the `John Smith` user account. John has a user ID of `john` and uses a plain text password of `mypassword`. Note that after configuration has been committed, only the encrypted version of the password is displayed when configuration is shown.

Note: User information can be changed through the UNIX shell (providing you have sufficient permission). However, any changes to router user accounts or authentication through the UNIX shell are overwritten the next time you commit router CLI configuration.

CAUTION: If your login user is not a member of the login user group "secrets" and you save a configuration either through the REST API or use the `save` command, the encrypted passwords in the configuration file are replaced with the ******** placeholder. If you load this configuration, the replaced password fields trigger validation errors because the placeholder does not match the format for an encrypted password. Do not commit this configuration. If you ignore the error message and perform a commit with this invalid configuration, the passwords are deleted.

To create a login user account, perform the following steps in configuration mode.

Table 31. Creating a login user account

Step	Command

Create the user configuration node, define the user ID, and give the full name of the user.	`vyatta@R1#set system login`
	`user john full-name “John`
	` Smith”`

Specify the password for the user in plain text.	`vyatta@R1#set system login`
	` user john authentication`
	` plaintext-password mypassword`

Commit the changes. 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"`
	`}`

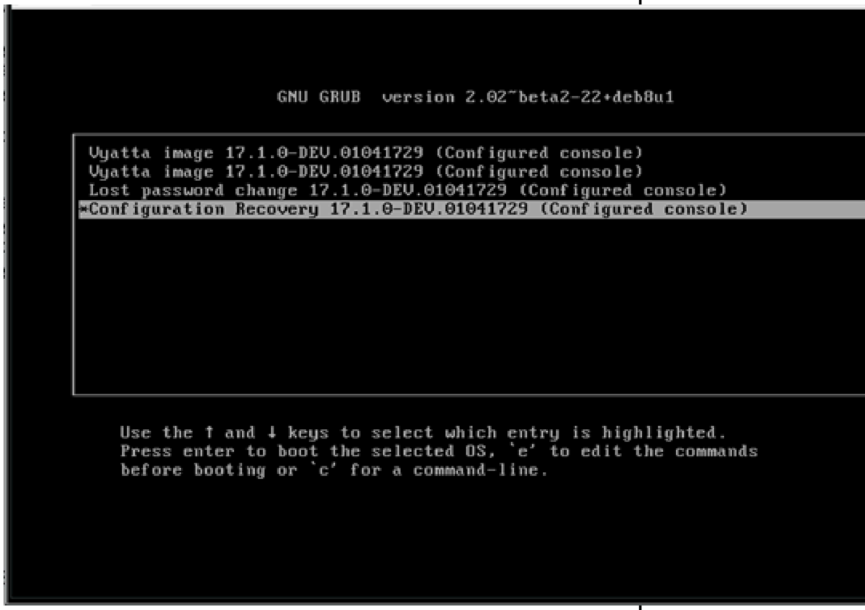
GRUB menu configuration options

You can use the GRUB menu configuration options on a router at boot time to recover system user configuration and passwords for local system users.

Recovering system user configuration

To recover system user configuration on a router, perform the following steps from a console window.

Table 32. Recovering system user configuration

<table>
<tr><th>Step</th><th>Command</th></tr>
<tr><td>
<p>Using the console, restart the router. The GRUB menu appears.</p>
<p>Select the <code>Configuration Recovery</code> option for the desired system software version.</p>
<p><b>Note:</b> Versions earlier than 17.1 do not support configuration recovery from the GRUB menu.</p>
</td><td>

<pre>
GNU GRUB version 2.02~beta2-22+deb8u1

 Vyatta image 17.1.0-DEV.01041729 (Configured console)
 Vyatta image 17.1.0-DEV.01041729 (Configured console)
 Lost password change 17.1.0-DEV.01041729 (Configured console)
*Configuration Recovery 17.1.0-DEV.01041729 (Configured console)

   Use the ↑ and ↓ keys to select which entry is highlighted.
   Press enter to boot the selected OS, `e' to edit the commands
   before booting or `c' for a command-line.
</pre>
</td></tr>
<tr><td>
<p>Enter the relevant option from the Configuration Recovery list and press <b>Enter</b>. The action of the selected option is performed.</p>
<p>Up to five previous committed versions are displayed on each screen. You can navigate to the next or previous committed versions by using the <code>Next</code> or <code>Previous</code> options.</p>
<p><b>Note:</b> The number of previous configurations displayed is set by the value of the system configuration management commit revisions. The default value is 20.</p>
<p>The commit configuration recovery menu also displays the commit comment to help you understand the details of the configurations.</p>
</td><td>
<pre>
 Configuration Recovery
======================

P   Previous
5   2017-01-04 20:54:29 by vyatta
6   2017-01-04 20:48:10 by root
         Restored '2017-01-04 20:46:28' commit following reboot
7   2017-01-04 20:46:28 by vyatta
8   2017-01-04 20:31:55 by root
         Restored '2017-01-04 20:29:35' commit following reboot
9   2017-01-04 20:29:35 by vyatta justTesting
N   Next
Q   Quit and reboot

Please choose option: 8

Restoring configuration version 8 ...

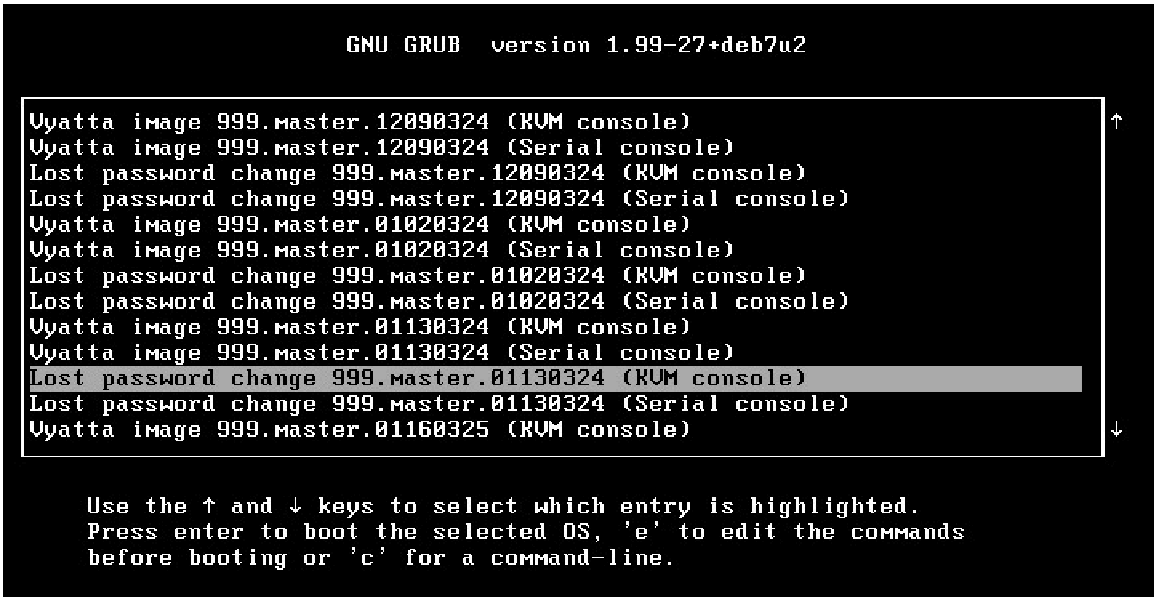
System will reboot in 10 seconds...
</pre>
</td></tr>
</table>

Recovering a system user password

To recover a system user password on a router, perform the following steps from a console window.

Table 33. Recovering a system user password

Step	Command

Using the console, restart the router. The GRUB menu appears.
Select the relevant option from the GRUB menu and press **Enter**. The option must start with “Lost password change.”
The standalone user-password recovery tool starts running and prompts you to reset the local system user password.
	
	`GNU GRUB  version 1.99-27+deb7u2`
	`Vyatta image 999.master.12090324 (KVM console)`
	`Vyatta image 999.master.12090324 (Serial console)`
	`Lost password change 999.master.12090324 (KVM console)`
	`Lost password change 999.master.12090324 (Serial console)`
	`Vyatta image 999.master.01020324 (KVM console)`
	`Vyatta image 999.master.01020324 (Serial console)`
	`Lost password change 999.master.01020324 (KVM console)`
	`Lost password change 999.master.01020324 (Serial console)`
	`Vyatta image 999.master.01130324 (KVM console)`
	`Vyatta image 999.master.01130324 (Serial console)`
	`Lost password change 999.master.01130324 (KVM console)`
	`Lost password change 999.master.01130324 (Serial console)`
	`Vyatta image 999.master.01160325 (KVM console)`
	`Use the ↑ and ↓ keys to select which entry is highlighted.`
	`Press enter to boot the selected OS, 'e' to edit the commands`
	`before booting or 'c' for a command-line.`

Enter **y** and follow the instructions to reset	`Standalone user password recovery tool.`
	`Do you wish to reset the local system user password? (y or n)`
	`y`
	
	`Starting process to reset the password...`
	
	`Re-mounting root filesystem read/write...`
	
	`Enter the local username for password reset:`

Table 33. Recovering a system user password (continued)

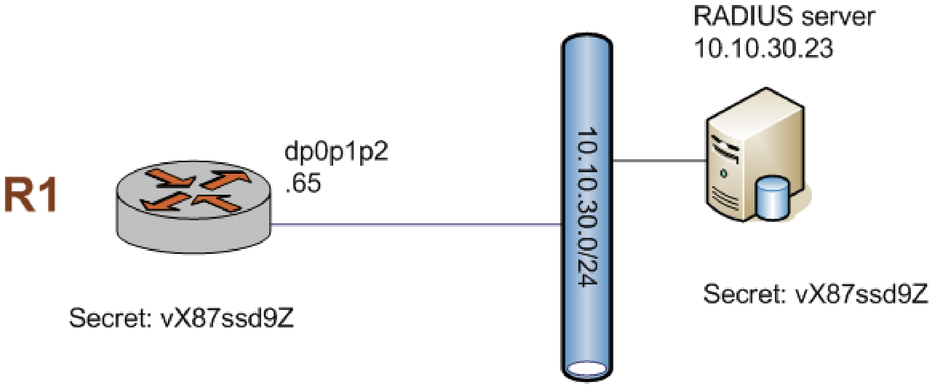
Step	Command

the password.
After the router starts, log in by using the new password.
	`vyatta`
	
	`Setting the user (vyatta) password...`
	`Enter vyatta password:`
	`Retype vyatta password:`
	`System will reboot in 10 seconds...`

Configuring a system for a RADIUS authentication server

This section provides a sample configuration of a router for a RADIUS authentication server, as shown in the following figure.

Figure 7. Configuration of a RADIUS authentication server



The example shows how to define a RADIUS authentication server at the 10.10.30.23 IP address. The system is to access the RADIUS server by 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: Carefully select the shared secret because this secret (string of characters) prevents snooping attacks on passwords. This secret, or key, is used on every packet, so it is important to choose a key that makes brute-force attacks more difficult; this key should be harder to guess than any password on the system.

To define this RADIUS authentication server, perform the following steps in configuration mode.

Table 34. Configuring a system for a RADIUS authentication 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`

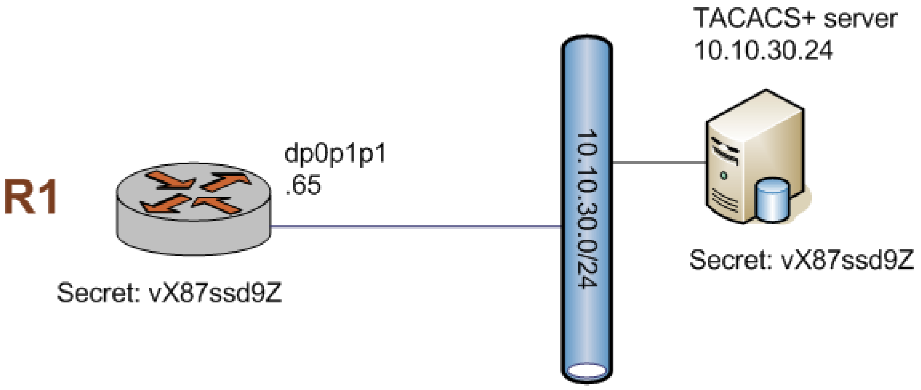
Save the configuration so that the changes persist after reboot.	`vyatta@R1# save`
	`Saving configuration to '/config/config.boot'...`
	`Done`

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 a system for a TACACS+ authentication server

This section provides a sample configuration of a router for a TACACS+ authentication server, as shown in the following figure.

Figure 8. Configuration of a TACACS+ authentication server



The example shows how to define a TACACS+ authentication server at the 10.10.30.24 IP address. The system is to access the TACACS+ server by 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. The default port is 49 and the default timeout is 3 seconds.

Note: Carefully select the shared secret because this secret (string of characters) prevents snooping attacks on passwords. This secret, or key, is used on every packet, so it is important to choose a key that makes brute-force attacks more difficult; this key should be harder to guess than any password on the system.

To define this TACACS+ authentication server, perform the following steps in configuration mode. Run `$ configure` to enter the configuration mode.

Table 35. Configuring a system for a TACACS+ authentication 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`

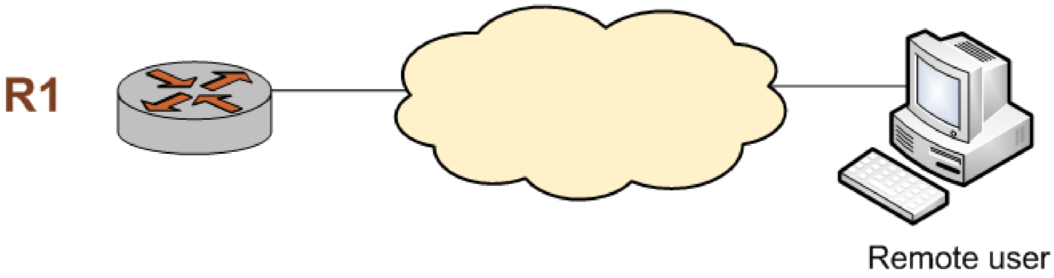
Show the contents of the **system tacplus-server** configuration node.	`vyatta@R1:~$ show system login tacplus-server`
	
	`tacplus-server 10.10.30.24 {`
	`    secret "********"`
	`}`

Show the status of TACACS+.	`vyatta@R1:~$ show system tacplus status`
	
	`Server address: 10.10.30.24 (active)`
	`Server port: 49`
	`Authentication requests/replies: 1/1`
	`Authorization requests/replies: 2/2`
	`Accounting requests/replies: 5/5`
	`Failed connects: 0`

Configuring a system for SSH access using shared public keys

This section provides a sample configuration of a router for SSH access by using shared public keys, as shown in the following figure.

Figure 9. Configuration for SSH access by using shared public keys



The example shows how to configure a router for SSH access that uses shared public keys for authentication and to disable password authentication (though disabling password authentication is not a prerequisite to using shared public keys for authentication). In this case, the `John Smith` user (username = `john`) already exists on the system. In addition, the public key (`xxx.pub`) was previously generated (by using the Linux `ssh-keygen` command) and is located in a directory owned by the `j2` user on `xyz.abc.com`.

To configure a system for SSH access by using shared public keys, perform the following steps in configuration mode.

Table 36. Configuring a system for SSH access by 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 to use only shared public key authentication.	`vyatta@R1# set service ssh disable-password-authentication`
Commit the change.	`vyatta@R1# commit`
Display the changes.	`vyatta@R1# show service ssh disable-password-authentication`

Load the shared public key (`xxx.pub`) from the system on which it is located and associate it with the user named `john`. In this case, it is located on `xyz.abc.com` in a directory owned by the `j2` user.	`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+/UjFR`
	`FrHbqHNhjxdlYviXveVXoZrKAKHtANRp5E+j4WZMbSd4oYt9P9lFevyZv3xmd`
	`ZE+ukuPlQBBAUnL29k1FtJ+G7I5tXGun9VR07JzUpEb8/KP1U4ajYClc3HxpO`
	`Lpu5AU5u7jvKu/wA0=`
	`             type ssh-rsa`
	`         }`
	`    }`
	`    full-name "John Smith"`
	`}`

Chapter 13. User Management Commands

loadkey

Loads a shared public key for a Secure Shell (SSH) user.

```
loadkey  userfile-name
```

user

The name of a user with which to associate a public key. The user must already be defined on the router.

file-name

The name of a shared public key file, including the full path to its location. A shared public key files is typically generated on the remote system by using the Linux `ssh-keygen` command and has a `.pub` extension. Its contents include the authentication type (for example, `ssh-rsa` or `ssh-dsa`), key, and remote system user ID (for example, name@domain.com).

Configuration mode

Use this command to load a shared public key for SSH from a file into the `public-keys` configuration for a user (refer to the system login user authentication public-keys. Loading a key from a file avoids having to manually enter the shared public key.

Note: This command can be run only 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 is overwritten.

The following table shows how to specify the syntax for files from different file locations.

Table 37. 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 username on the host, *passwd* is the password associated with the username, *host* is the host name or IP address of the FTP server, and *key-file* is the key file, including the path. If you do not specify *user* and *passwd*, you are prompted for them.
SCP server	Use the following syntax for *file-name*: scp://*user*:*passwd*@*host* /*key-file* where *user* is the username on the host, *passwd* is the password associated with the username, *host* is the host name or IP address of the SCP server, and *key-file* is the key file, including the path.

Table 37. Specifying locations for the shared public key file (continued)

Location	Specification
	If you do not specify *user* and *passwd* , you are prompted for them.
HTTP server	Use the following syntax for file-name: http://*host*/*key-file* 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 the login credentials of the current user.

```
show login  [ groups | level | user ]
```

Displays all credentials of the current user.

groups

Displays the groups to which the user belongs.

level

Displays the login level of the user.

user

Displays the login ID of the user.

Operational mode

Use this command to display the login credentials of the current user.

The following example shows how to display the login credentials of 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 vyattacfg
vyatta@R1:~$
```

show system login users

Displays information about user accounts.

```
show system login users  [ all | locked | other | vyatta ]
```

Displays information about router accounts.

all

Displays information about all accounts.

locked

Displays information about locked accounts.

other

Displays information about non-router accounts.

vyatta

Displays information about router accounts.

Operational mode

Use this command to display information about system accounts including information about the last time each user logged in.

The following example shows how to display information about router user accounts on R1.

```
vyatta@vyatta#  show system login user
user vyatta {
     authentication {
         encrypted-password $1$4XHPj9eT$G3ww9B/pYDLSXC8YVvazP0
     }
     level admin
 }
```

show system tacplus status

Displays the status of TACACS+.

```
show system tacplus status
```

Operational mode

Use this command to display the status of TACACS+.

The following example shows how to display the status of TACACS+. In this example, the (active) label, which appears next to a server address, has no bearing on the success of the attempted connection. This label identifies the last TACACS+ server to which the router tried to connect.

```
vyatta@vyatta:~$ show system tacplus status
Server address: 192.168.122.7 (active)
Server port: 49
Authentication requests/replies: 1/1
```

```
Authorization requests/replies: 2/2
Accounting requests/replies: 5/5
Failed connects: 0

Server address: 192.168.122.6
Server port: 60
Authentication requests/replies: 0/0
Authorization requests/replies: 0/0
Accounting requests/replies: 0/0
Failed connects: 1
```

The following example shows the message that is displayed if TACACS+ is not configured.

```
vyatta@vyatta:~$ show system tacplus status

  |
Tacplus daemon is not running.
```

The following example shows that the TACACS+ server at 1.1.1.1 runs in the VRF named red.

```
vyatta@vyatta:~$ show system tacplus status

  |
VRF red
Server address: 1.1.1.1 (active)
Server port: 49
Authentication requests/replies: 1/1
Authorization requests/replies: 2/2
Accounting requests/replies: 4/4
Failed connects: 0
```

system login

Creates the configuration node for user management and authentication.

`set system login`

`delete system login`

`show system login`

Configuration mode

```
system {
    login {
    }
}
```

Use this command to create the configuration node for user management 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 and authentication information.

Use the `show` form of this command to display user and authentication configuration.

system login banner post-login

Creates the text of the post-login banner.

```
set system login banner post-login  banner
```

```
delete system login banner post-login
```

```
show system login banner post-login
```

The system displays information about the operating system and copyright.

banner

The text (banner) to be displayed during login after a user enters a valid password. The banner must be enclosed in double quotation marks (“). Special characters such as new line (\n) and tab (\t) can also be entered.

Configuration mode

```
system {
    login {
        banner {
            post-login banner
        }
    }
}
```

Use this command to create the text (banner) that appears when a user logs in to the system successfully.

Use the `set` form of this command to create the post-login banner.

Use the `delete` form of this command to return to the default post-login banner, which is information about the operating system and copyright.

Use the `show` form of this command to display the post-login banner.

system login banner pre-login

Create the text of the pre-login banner.

```
set system login banner pre-login  banner
```

```
delete system login banner pre-login
```

```
show system login banner pre-login
```

The system displays a welcome message.

banner

The text (banner) to be displayed during login after a user enters a login ID. The banner must be enclosed in double quotation marks (“). Special characters such as new line (\n) and tab (\t) can also be entered.

Configuration mode

```
system {
    login {
        banner {
            pre-login banner
        }
    }
}
```

Use this command to create the text (banner) that appears when a user enters a login ID.

Use the `set` form of this command to create the pre-login banner.

Use the `delete` form of this command to return to the default pre-login banner which is a welcome message.

Use the `show` form of this command to display the pre-login banner.

system login group

Specifies the text of the group name.

```
set system login group  group-name
```

```
delete system login group  group-name
```

```
show system login group
```

The system displays a welcome message.

group

The group to be named.

Configuration mode

```
system {
    login {
        group group-name
    }
}
```

Use the `set` form of this command to create the group name.

Use the `delete` form of this command to delete the group name.

Use the `show` form of this command to display the group name.

system login radius-server

Defines a Remote Authentication Dial-In User Service (RADIUS) server for user authentication.

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**]

address

Multinode. The IP address of a remote authentication server running the RADIUS protocol. This server authenticates multiple users.

You can define multiple RADIUS servers by creating multiple `radius-server` configuration nodes.

port

Optional. A port to be used for RADIUS traffic. The default port is 1812.

secret

The secret (password) for the RADIUS server. This secret must be the same as that recorded on the RADIUS server.

The secret consists of alphanumeric and printable special characters (for example, the space character is not permitted). The secret is case sensitive.

timeout

Optional. The time-out (interval), in seconds, after which, if the RADIUS server has not responded, the next configured RADIUS server should be queried. The time-out ranges from 1 through 30. The default time-out is 2.

Configuration mode

```
system {
     login {
          radius-server address {
               port port
          }
     }
}
```

Use this command to define a RADIUS server and specify the information necessary to log in to it.

The RADIUS secret is specified and stored in plain text on the system and is used as part of a cryptographic operation for transferring authentication information securely over the network. When you view a RADIUS secret, it is 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 display RADIUS server configuration.

system login session-timeout

Defines system idle session timeout value in seconds.

```
set system login session-timeout  { 0 | 0-4294967295 }
```

```
delete system login session-timeout  [ 0 | 0-4294967295 ]
```

```
show system login session-timeout
```

Disabled.

0

Disables session time out.

0-4294967295

Session idle duration in seconds before timeout.

Configuration mode

```
system {
     login {
          session-timeout value
     }
}
```

Use the `set` form of this command to define the system idle session timeout value in seconds.

Use the `delete` form of this command to remove the system idle session timeout value and to restore the default configuration.

Use the `show` form of this command to display the system idle session timeout value.

system login tacplus-server

Defines a Terminal Access Controller Access Control System Plus (TACACS+) server for user authentication.

```
set [ routing routing-instance vrf-name ] system login tacplus-server  address [
port port | secret secret | source-address source-address| source-interface
source-interface | timeout timeout ]
```

```
delete [ routing routing-instance vrf-name ] system login tacplus-server  address
[ port | secret | source-address| source-interface source-interface | timeout
]
```

```
show [ routing routing-instance vrf-name ] show system login tacplus-server
address  [ port | secret | source-address| source-interface source-interface |
timeout ]
```

vrf-name

The name of the VRF instance for which this command is configured.

address

Multinode. The IP address or host name of a remote authentication server running TACACS +. This server authenticates multiple users.

You can define multiple TACACS+ servers by creating multiple `tacplus-server` configuration nodes. Multiple servers are prioritized in the order in which they are configured.

port

A port to use for TACACS+ traffic. The default port is 49.

secret

The secret (password) for the TACACS+ server. This secret must be the same as that recorded on the TACACS+ server.

The secret consists of alphanumeric and printable special characters (for example, the space character is not permitted). The secret is case sensitive.

source-address

An IP address to use as the source address when connecting to the TACACS+ server. This address is typically not required.

source-interface

The source interface to use for connecting to the server.

timeout

The time-out (interval) in seconds after which, if the TACACS+ server has not responded, the next configured TACACS+ server should be queried. The time-out range is 1-30. The default time-out is 3.

Configuration mode

```
routing {
    routing-instance {
        system {
             login {
                  tacplus-server {
                       port
                       secret
                       source-address
                       source-interface
                       timeout timeout
                  }
             }
        }

    }
}
```

Use this command to define a TACACS+ server and specify the information necessary to log in to it.

The TACACS+ secret is specified in plain text and stored in plain text on the system and is used as part of a cryptographic operation for transferring authentication information securely over the network. When you view a TACACS+ secret, it is displayed in plain text.

Note: TACACS+ servers are not supported for IPv6.

Users doing packet capture 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 display TACACS+ server configuration.

system login user

Creates a user account.

```
set system login user  user
```

```
delete system login user  user
```

```
show system login user  user
```

user

Multinode. 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.

Configuration mode

```
system {
     login {
          user user
     }
}
```

Use this command to define a user that is authenticated by using the internal mechanism of the system: “login” authentication.

Note that, although user account and authentication information can be changed by using the operating system shell, the system overwrites these changes the next time you commit configuration in the Vyatta shell. For persistent changes to user or authentication information, use Vyatta CLI commands.

In addition, 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 display `user` configuration.

system login user authentication

Sets an authentication password for a user.

`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`** `]`

user

A user ID.

epwd

The encrypted password. This password consists of the encrypted characters of the actual password. You can obtain the encrypted characters of the actual password by using the `mkpasswd` command on the VM.

ppwd

The password for the user, specified in plain text. Most special characters can be used with the exception of single quotation marks ('), double quotation marks (“), and backslashes (\).

Configuration mode

```
system {
    login {
        user user {
            authentication {
            encrypted-password epwd
            plaintext-password ppwd
            }
        }
    }
}
```

Use this command to set a password to authenticate a user. When the encrypted password is displayed, the encrypted value is shown. The plain text password appears as double quotation marks in the configuration.

⚠ **CAUTION:** If your login user is not a member of the login user group "secrets" and you save a configuration either through the REST API or use the `save` command, the encrypted passwords in the configuration file are replaced with the ******** placeholder. If you load this configuration, the replaced password fields trigger validation errors because the placeholder does not match the format for an encrypted password. Do not commit this configuration. If you ignore the error message and perform a commit with this invalid configuration, the passwords are deleted.

To disable a user account without deleting it, you can simply set the value of the `encrypted-password` option to an asterisk (*).

Use the `set` form of this command to set the password for a user.

Use the `delete` form of this command to remove the password for a user.

Use the `show` form of this command to display user password configuration.

system login auth-chain method

Sets the order of the authentication.

```
set system login auth-chain  [ method tacplus | method local ]
```

```
delete system login auth-chain  [ method tacplus | method local ]
```

```
show system login
```

The default order for the authentication method is TACAS+ server followed by local system-user login.

`auth-chain {` **method** `tacplus;` **method** `login}`

method tacplus

Specifies the authentication method as TACACS+ server.

method local

Specifies the authentication method as local system-user login.

Configuration mode

```
system {
    login {
        auth-chain {
            method tacplus
            method local
        }
    }
}
```

Use this command to set the order of authentication by using the authentication chaining method. The system performs authentication in the order of the authentication chain. The scenarios for authentication chaining follow.

- If you specify the authentication method as local, the system uses the local system-user login to authenticate.
- If you specify the authentication method as TACACS +, the system uses the TACACS + authentication. The authentication chain does not proceed to use the local authentication unless the TACAS+ authentication is configured but not working.
- If you use both the TACACS + and local authentication methods, the system attempts the first method. If the first method is successful, the chain does not proceed. If the first method fails, the authentication chain proceeds and the system attempts the next method.

Use the `set` form of this command to set the order of the authentication chain.

Use the `delete` form of this command to remove the order of the authentication chain.

Use the `show` form of this command to display the order of the authentication chain.

system login user authentication public-keys

Specifies parameters for a Secure Shell (SSH) shared public key user authentication.

`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**]

user

A user ID.

key-id

A key identifier. This identifier 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.

key-options

Additional options separated by commas. 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 key (authentication) type to be used, which must be specified. They key is either of the following:

`ssh-dsa`—Specifies DSA authentication.

`ssh-rsa`—Specifies RSA authentication.

Configuration mode

```
system {
    login {
        user user {
            authentication {
                public-keys key-id {
                    key key-value
                    options key-options
                    type key-type
                }
            }
        }
    }
}
```

Use this command to specify the parameters to be used for shared public key authentication for logins by using SSH. During commit, these values are placed in the `/home/<user>/.ssh/authorized_keys` file. Changes to this file can be made only by using this command. All direct user changes to this file are lost.

Rather than specifying these parameters directly by using the `set` form of this command, the recommended method is to use the loadkey. It populates 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 appears if a shared public key is not provided at the start of the SSH session. 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, refer to *Services Configuration 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 display public key parameters.

system login user full-name

Records the full name of a user.

```
set system login user  user full-name name
```

```
delete system login user  user full-name
```

```
show system login user  user full-name
```

user

A user ID.

name

A character string that represents the name of the user, including alphanumeric characters, space, and hyphens (-). A character string that includes spaces must be enclosed in double quotation marks (“).

Configuration mode

```
system {
    login {
        user user {
            full-name name
        }
    }
}
```

Use this command to record the full name of a user.

Use the `set` form of this command to specify the name of a user.

Use the `delete` form of this command to remove the name of a user.

Use the `show` form of this command to display the name of a user.

system login user group

Assigns a user to a group.

`set system login user` *user* **group** *group*

`delete system login user` *user* **group**

`show system login user` *user* **group**

user

A user ID.

group

A character string that represents the group to which the user is to be assigned. Groups are defined in the `/etc/group` directory.

Configuration mode

```
system {
    login {
        user user {
             group group
        }
    }
}
```

Use this command to assign a user to a group. A user can be a member of multiple groups by running this command once for each group to which the user is to be assigned.

Use the `set` form of this command to make a user a member of a group.

Use the `delete` form of this command to remove a user from a group.

Use the `show` form of this command to display the groups to which a user is assigned.

system login user home-directory

Specifies the home directory of a user.

`set system login user` *user* **home-directory** *dir*

`delete system login user` *user* **home-directory**

`show system login user` *user* **home-directory**

The home directory is `/home/`*user*.

user

A user ID.

dir

A character string that represents the home directory of the user. The following is an example: `/home/vyatta`

Configuration mode

```
system {
     login {
          user user {
               home-directory dir
          }
     }
}
```

Use this command to specify the home directory of a user.

Use the `set` form of this command to specify the home directory of a user.

Use the `delete` form of this command to restore the default home directory of a user, which is **/home/***user*.

Use the `show` form of this command to display the home directory of a user.

system login user level

Specifies the privilege level and system access of a user.

```
set system login user  user level level
```

```
delete system login user  user level
```

```
show system login
```

A user is assigned administrative privileges.

user

A user ID.

level

The privilege level of the user. The level is either of the following:

`admin`—Assigns administrative privilege to the user. The user can run any command in the Vyatta CLI or the underlying operating system.

`operator`—Assigns restricted privilege to the user. The user can run operational commands in the Vyatta CLI plus restricted forms of the `ping` and `traceroute` commands. The user cannot enter configuration mode or run configuration commands.

`superuser`—A superuser has the privilege of an **admin** user. In addition to that, a superuser has access to install or update additional packages, and access or modify internal system files and so on.

Configuration mode

```
system {
    login {
        user user {
            level level
        }
    }
}
```

Use this command to assign role-based system access to a user.

The system supports two system roles:

- Administrator (admin): A user that is assigned a role of admin has full access to all Vyatta-specific commands plus all operating system shell commands. Access to operating system shell commands is direct: the user does not need exit to another shell mode before running these commands. Although admin users can run any command implemented in the system, command completion and CLI help show only router commands.
- Operator: A user that is assigned a role of operator has access to the router operational command set but no access to configuration commands. An operator also has limited access to operating system commands. At this time, command completion and CLI help show all router commands for a user with the operator role.

Use the `set` form of this command to assign the privilege level to a user.

Use the `delete` form of this command to restore the privilege level of a user to the default level, which is administrative level.

Use the `show` form of this command to display the privilege level of a user.

system tacplus-options command-accounting

Enables logging of accounting records for interactive shell (vbash) commands.

```
set system tacplus-options command-accounting
```

```
delete system tacplus-options command-accounting
```

```
show system tacplus-options
```

Accounting records are not logged.

Configuration mode

```
system {
     tacplus-options {
          command-accounting
     }
}
```

Use this command to enable logging of accounting records for interactive shell commands.

Connections to the system for which commands are logged include SSH, Telnet, console, and serial. Command logging is not limited to TACACS+ authenticated users and accounts for interactive shell commands. Accounting records are logged to the TACACS+ server.

Use the `set` form of this command to enable logging of accounting records for interactive shell commands.

Use the `delete` form of this command to restore the default behavior for command accounting, that is, accounting records are not logged.

Use the `show` form of this command to display the configuration of command accounting.

Chapter 14. Service-user Management

This chapter describes service-user management on the router.

Overview

Service-user management handles authentication for services and is not intended to be used to access the router for administrative purposes. The administration of service-user management is done at the system-login configuration level.

This chapter describes service-user management, which is controlled at the `resources service-users` configuration level. Configuration is set in a central location within the `resources service-users` configuration section.

Other services that require service-user authentication, such as OpenVPN, refer to authentication profiles, or group of users, in the `resource service-users` section.

The router allows you to connect to existing Lightweight Directory Access Protocol (LDAP) services in your organization for authentication purposes and maintain a local user database that does not require any pre-existing identity service in your environment.

All changes for service users do not require any service interruption or service restart.

Note: Service-user management includes revoking access or deleting user accounts, which does not terminate an existing service-user session of services.

All service users are granted access to the Service-User Web Portal, which is available at the following address:

URL: https://<IP address of router>/service

To enable this portal, use the following command:

```
vyatta@vyatta# set service https service-users
```

Local service user

This section covers how to grant and allow access to services to a local service user.

A local service user is maintained at the local service-user configuration level under `resources service-users`.

Authentication for a local service user is gained by using a username and password. The password in the CLI is provided as a plain-text password or as an encrypted SHA-512 hash. A plain-text password is stored as an SHA-512 hash after the configuration is committed.

Setting a username and password

You must set a username and an authentication password to create the minimum configuration that is required of a local service user.

To create the alice user with the password of secretpw, use the following command:

```
vyatta@vyatta# set resources service-users local user alice auth
 plaintext-password secretpw
```

Granting service access to a user

Setting a username and password does not grant the alice user access to service. Users are not granted access to any service by default.

To grant access to OpenVPN for alice, you must give alice access to the vtun0 tunnel interface by entering the following command:

```
vyatta@vyatta#set openvpn vtun0 auth local user alice
```

The alice user is now granted access to vtun0 and authenticated by the specified username and password.

Note: For OpenVPN, additional settings are required to allow authentication by the username and password. For details, see the “SSL-VPN Client Bundler” section in *OpenVPN Configuration Guide*.

In general, granting or revoking access to any service does not require a restart of the service—that is, a service interruption.

Revoking service access for a user

To revoke access to vtun0 for the alice user, use the following command:

```
vyatta@vyatta# delete openvpn vtun0 auth local user alice
```

Note: By revoking access to vtun0, the existing service session (for example, an OpenVPN tunnel connection) is not interrupted or terminated—termination must be done manually.

Locking services from a user

Alternatively, a local service user can be temporarily locked from all services on the router by locking the user.

To lock a user, use the following command:

```
vyatta@vyatta# set resources service-users local user alice lock
```

Note: Revoking access to individual services does not interrupt or terminate an existing service session.

Unlocking services from a user

To remove the service-user lock, use the following command:

```
vyatta@vyatta# delete resources service-users local user alice lock
```

Granting access service to a group

To maintain a larger set of local users, you must group the users and reference the service configuration to which the group of users should be granted access.

To grant access to an OpenVPN endpoint that is dedicated for use by the sales department to the alice and bob local service users, both of whom work in the sales department, use the following commands:

```
vyatta@vyatta# set resources service-users local group sales-dep

vyatta@vyatta# set resources service-users local user alice group sales-dep

vyatta@vyatta# set resources service-users local user bob group sales-dep
```

To grant the sales-dep group access to the OpenVPN vtun1 interface, use the following command:

```
vyatta@vyatta# set interfaces openvpn vtun1 auth local group sales-dep
```

Service-user authentication through LDAP

To create an LDAP profile to allow authentication against an existing LDAP service in your network, the following are required:

- Existing network connection or a route to the LDAP server
- LDAP server that is configured for Transport Layer Security (TLS) with StartTLS or LDAP over Secure Sockets Layer (SSL) (ldaps://)

Note: Encryption is required for the exchange of the authentication token.

Creating an LDAP authentication profile

To create an LDAP authentication profile, configured with minimum settings, the following are required:

- Authentication that is granted against the Example corporate LDAP server, which can be reached through the fully qualified domain name (FQDN) of ldap.example.com
- Authentication that is configured with TLS and supports StartTLS

To configure the LDAP server URL with StartTLS `ldap://` (for LDAP+SSL: ldaps:), use the following command:

```
vyatta@vyatta# set resources service-users ldap example.com url
 ldap://ldap.example.com
```

If a custom port is required, the port can be specified in the URL by appending the port number to the FQDN; for example: `ldap://ldap.example.com:1234`.

The default FQDN ports, according to a generally accepted standard, are as follows if not otherwise specified.

Table 38. Default ports for FQDN

FQDN	Port Number
ldap://	389
ldaps://	636

Setting the base distinguished name

To set the Base Distinguished Name (Base DN) of an LDAP v3 server for an organization that is used for authorization, use the following command:

```
vyatta@vyatta# set resources service-users ldap example.com base-dn
 ou=People,dc=example,dc=com
```

Applying the LDAP search filter to an LDAP entry

To apply the LDAP search filter to each LDAP entry that matches the username and LDAP member attribute, use the following command:

```
vyatta@vyatta# set resources service-users ldap example.com search-filter
 (objectClass=posixAccount)
```

The LDAP search filter also supports more-complex search filters, as described in RFC2254.

The following are the minimum required attributes that must be set for the LDAP authentication of service users.

- URL

- Base-dn
- Search-filter

Configuring the bind user and bind password

If the LDAP server does not allow anonymous binding, an LDAP bind user and bind password must be configured by using the following commands:

```
vyatta@vyatta# set resources service-users ldap example.com bind-dn
bind-username

vyatta@vyatta# set resources service-users ldap example.com password
 bindpw
```

Specifying a trusted CA certificate

If the TLS or SSL certificate that is issued by a corporate certificate authority (CA) is not trusted or known to the router, the required certificate must be explicitly specified.

To specify this certificate, use the following command:

```
vyatta@vyatta# set resources service-users ldap example.com tls
 cacert /config/auth/ldap-ca.pem
```

Alternatively, to reduce the number of checks on the TLS or SSL LDAP server certificate, use the following command:

```
vyatta@vyatta# set resources service-users ldap example.com tls reqcert
 {never | allow | try | demand}
```

If no option is explicitly specified, the `demand` option is set by default.

Table 39. Variable definitions

Option	Description
`never`	Performs no request and no checks on the server certificate.
`allow`	Requests and checks the certificate, if available. Tolerates bad server certificates.
`try`	Requests and checks the certificate, if available. Bad server certificates get rejected.
`demand`	Requests a valid server certificate (default).

Gaining authentication from multiple LDAP servers

To gain authentication for a service from multiple different LDAP servers and LDAP trees, you must create two different LDAP authentication profiles by using the following commands:

```
vyatta@vyatta# set resources auth ldap example.com url
 ldap://ldap.example.com

vyatta@vyatta# set resources auth ldap example.com ...

vyatta@vyatta# set resources auth ldap emea.example.com url
 ldap://ldap.emea.example.com

vyatta@vyatta# set resources auth ldap emea.example.com ...
```

To specify both LDAP profiles in the configuration of a service authentication, use the following commands:

```
vyatta@vyatta# set interfaces openvpn vtunX auth ldap example.com

vyatta@vyatta# set interfaces openvpn vtunX auth ldap emea.example.com
```

When a service user tries to authenticate the OpenVPN vtunX interface, the provided credentials are authenticated against all the provided LDAP profiles.

A single access-granting LDAP profile is sufficient for the service user to successfully establish the OpenVPN connection. Access is not required to be granted by all the configured LDAP profiles.

Note: The OpenVPN service authentication could be mixed with LDAP authentication profiles, local service users, or groups of local-service users.

To allow SSL-VPN clients to connect without a TLS client certificate that is specific to an end user, you must set the client-cert-not-required option. Even if client certificates were created, they are not included in any SSL-VPN client bundles.

```
# set interfaces openvpn vtunX client-cert-not-required
```

Performing group-based LDAP authorization

If the LDAP search filter is configured to perform a group-based LDAP authorization, you might need to restrict (that is, adapt) the search base to search for groups.

To adjust the search base for groups, use the following command:

```
     vyatta@vyatta# set resources service-users ldap example.com group
base-dn ou=Groups,dc=example,dc=com
```

Depending on the defined LDAP schema (RFC2307 or RFC2307bis), the member attribute is either `memberuid` or `member` for the group-based authentication.

If the LDAP schema used by the server requires a third variant that is not covered by either schema standard, use the following command:

```
      vyatta@vyatta# set resources service-users ldap example.com group
 member-attribute
 memberAttr
```

Setting advanced LDAP options

LDAP referrals are not used by the LDAP server by default.

To configure the server to follow LDAP referrals, use the following command:

```
vyatta@vyatta# set resources service-users ldap example.com
 follow-referrals
```

LDAP service-user management supports two LDAP schema standards: RFC2307 and RFC2307bis. The main difference between the two standards is how the member attribute of groups is stored.

According to RFC2307, the members of a group are stored in the LDAP attribute `memberuid`. According to RFC2307bis, the members of a group are stored in `member`. These settings depend on the LDAP schema that is used on the LDAP server.

To set the RFC2307bis schema standard as the default, use the following command:

```
vyatta@vyatta# set resources service-users ldap example.com schema
 rfc2307bis
```

Chapter 15. IPv6

This chapter describes commands for enabling IPv6 functionality on the system.

IPv6 overview

The router includes extensive support of IPv6. An overview of that support is available in *IPv6 Support Configuration Guide*.

IPv6 configuration

Examples of configuring basic IPv6 functionality are located in *IPv6 Support Configuration Guide*.

Chapter 16. IPv6 System Commands

reset ipv6 neighbors address

Removes an IPv6 address from the IPv6 Neighbor Discovery (ND) cache.

```
reset ipv6 neighbors address  ipv6
```

ipv6

An IPv6 address.

Operational mode

Use this command to remove an IPv6 address from the ND cache.

reset ipv6 neighbors interface

Removes an interface from the IPv6 Neighbor Discovery (ND) cache.

```
reset ipv6 neighbors interface  interface_name
```

interface_name

The identifier of an interface. Supported interface types are:

- Data plane
- Loopback

For more information about these interface types, refer to Loopback and Data Plane Interfaces.

Operational mode

Use this command to remove an Ethernet interface from the IPv6 ND cache.

show ipv6 neighbors

Displays the IPv6 Neighbor Discovery (ND) cache.

```
show ipv6 neighbors
```

Operational mode

Use this command to display the IPv6 ND cache.

Table 40: ND states shows possible ND states.

Table 40. ND states

State	Description
incomplete	Address resolution is currently being performed on this neighbor entry. A neighbor solicitation message has been sent, but a reply has not yet been received.
reachable	Address resolution has determined that the neighbor is reachable. Positive confirmation has been received, and the path to this neighbor is operationable.
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 router is waiting for a response from this neighbor.
failed	Neighbor reachability state detection failed.
noarp	The neighbor entry is valid. There are no attempts to validate it, but the neighbor can be removed from the cache when its lifetime expires.
permanent	The neighbor entry is valid indefinitely and should not be cleared from the cache.
none	No state is defined.

system ipv6 disable

Disables the assignment of IPv6 addresses on all interfaces.

```
set system ipv6 disable
```

```
delete system ipv6 disable
```

```
show system ipv6 disable
```

IPv6 addresses are assigned on all interfaces.

Configuration mode

```
system {
    ipv6 {
        disable
    }
}
```

Use this command to disable the assignment of IPv6 addresses on all interfaces.

Use the `set` form of this command to disable IPv6 address assignment on all interfaces.

Use the `delete` form of this command to enable IPv6 address assignment on all interfaces.

Use the `show` form of this command to display IPv6 disabling configuration.

system ipv6 disable-forwarding

Disables IPv6 forwarding on all interfaces.

```
set system ipv6 disable-forwarding
```

```
delete system ipv6 disable-forwarding
```

```
show system ipv6 disable-forwarding
```

IPv6 packets are forwarded.

Configuration mode

```
system {
    ipv6 {
        disable-forwarding
    }
}
```

Use this command to disable IPv6 forwarding on all interfaces. IPv6 forwarding can also be disabled for each interface by using the `ipv6 disable-forwarding` command associated with the interface (for example, `interfaces dataplane dp0p1p1 ipv6 disable-forwarding`). These commands are documented in the guides that describe the individual interfaces. For example, Ethernet interface commands are described in *LAN Interfaces Configuration Guide*.

Use the `set` form of this command to disable IPv6 packet forwarding on all interfaces.

Use the `delete` form of this command to enable IPv6 packet forwarding on all interfaces.

Use the `show` form of this command to display IPv6 packet forwarding configuration.

system ipv6 strict-dad

Disables IPv6 operation on an interface when Duplicate Address Detection (DAD) fails for a link-local address.

```
set system ipv6 strict-dad
```

```
delete system ipv6 strict-dad
```

```
show system ipv6 strict-dad
```

IPv6 operation is not disabled on an interface when DAD fails for a link-local address.

Configuration mode

```
system {
    ipv6 {
        strict-dad
    }
}
```

Use this command to disable IPv6 operation on an interface when DAD fails for a link-local address.

Link-local addresses are formed from an interface identifier that is partly derived from the hardware address of a device, which is assumed to be uniquely assigned.

By default, the duplicate address is not assigned to the interface, but IPv6 continues to operate. This command disables IPv6 on the interface when a duplicate of the link-local address is detected.

Use the `set` form of this command to disable IPv6 operation on an interface when DAD fails for a link-local address.

Use the `delete` form of this command to leave IPv6 operational on an interface when DAD fails for a link-local address.

Use the `show` form of this command to display DAD failure configuration.

Chapter 17. Hot-plugging Interfaces

Overview

A router supports *hot-plugging*, which allows a running router to automatically discover a PCI network interface that is virtually plugged into the router, that is, a guest virtual machine (VM), without having to restart the router. After the interface is discovered, you can configure it as a data plane interface as described in *LAN Interfaces Reference Guide*.

router hot-plugging is supported on the VMware ESX and Linux Kernel-based Virtual Machine (KVM) virtualization platforms.

Note: Ubuntu 14.04 comes with Linux kernel version 3.13.0, which does not support hot-plugging. To get hot-plugging to work on Ubuntu 14.04, you must upgrade your Ubuntu software to use Linux kernel version 3.13.1.

How hot-plugging works on the VMware ESX platform

Hot-plugging an interface into a router that runs in a VMware ESX host VM is automatic.

When you add a network interface to a running router by using VMware vSphere Client, the router automatically detects the interface and registers it with the kernel. Similarly, when you delete an interface from a running router, it unregisters the interface with the kernel.

Note: On a VMware ESX platform, as many as 10 interfaces can be hot-plugged into a router. The following table lists the names that are assigned to hot-plugged interfaces.

Table 41. Interface names

Slot	Interface Name
1	dp0p160p1
2	dp0p192p1
3	dp0p224p1
4	dp0p256p1
5	dp0p161p1
6	dp0p193p1
7	dp0p225p1
8	dp0p257p1
9	dp0p162p1
10	dp0p194p1

PCI slot assignment

When you hot-plug a network interface into a router, the router assigns the first virtualized PCI slot that is available to the interface. For example, if the first, second, and third PCI slots are in use, the new interface is hot-plugged into the fourth slot. In this case, the name of the new interface is dp0p256p1. However, if the interface that is associated with the second slot is deleted later, when you hot-plug a new interface, the new interface is plugged into the second slot (dp0p192p1).

Persistence

On the VMware ESX virtualization platform, by default, hot-plugged network interfaces persist through router restarts.

How hot-plugging works on the Linux KVM platform

On the KVM platform, to create and hot-plug an interface into a router, you can use the following command on the host KVM system.

virsh attach-device <vm-name> [**--persistent**] **<xml-filename>**

To detach interfaces from a running router, you can use the following command.

virsh detach-device <vm-name> [**--persistent**] **<xml-filename>**

Note: On a Linux KVM platform, as many as 32 interfaces can be hot-plugged into a router.

Note: The virsh tool is available through the libvirt toolkit.

PCI slot assignment

On the KVM platform, when hot-plugging a network interface into a router, unless you explicitly specify the PCI slot address, the router plugs the interface into the next available PCI slot with the higher slot number.

Example

If the first and third slots are in use, and a new interface is introduced, the new interface will take the next available PCI slot with the higher slot number. In this case, it will take PCI slot-4 and not PCI slot-2. The reason is that slot-2 is unavailable because it was previously used after the system was booted.

Persistence

On the Linux KVM virtualization platform, by default, hot-plugged network interfaces do not persist through router restarts. To ensure persistence, you can run the following command on the host KVM system:

virsh attach-device <vm-name> --persistent <xml-filename>

Naming of interfaces

During the boot sequence of a router, the VM assigns PCI slots in the order that interfaces are discovered.

To avoid the renaming of interfaces, either ensure that no temporary hot-plugged interfaces exist before the persistent interfaces or use the `<address .../>` clause in the XML file that is associated with the interface to specify the PCI slot into which to plug the interface.

Persistence of interface configurations

You can detach a network interface from a router by using the **virsh detach-device <vm-name>** [**--persistent**] **<xml-filename>** command. However, because the router configuration is independent of hot-plugging, when detaching an interface, the configuration that is associated with this interface remains in the config/config.boot file on the router.

If you attach an interface into the PCI slot that corresponds to the name of an existing interface in the config/config.boot file, then the router reuses the existing configuration by interface name.

Note: When you detach an interface, the router resets all MIB counters.

Hot-plugging Interfaces on the VMware ESX Platform

On the VMware ESX platform, perform the following steps to hot-plug a network interface into a router:

1. Log in to vSphere Client.
2. Add an Ethernet network adapter to your router and set the adapter type to VMXNET 3.

   Note: VMXNET 3 is the only supported network adapter type for hot-plugging on the VMware ESX platform.

The router hot-plugs the new interface.

Hot-plugging interfaces on the KVM platform

On the KVM virtualization platform, perform the following steps to hot-plug a network interface into a router:

1. Add an Ethernet network interface.
2. On the host KVM system, create an XML file that specifies the following information:
    - Interface type.
    - MAC address of the interface. You must ensure that this address is unique.
    - Label or name of the network device to which your router is connected.
    - (Optional) Virtualized PCI slot to which the interface is plugged.
    - Model type.

   For more information about the contents of the XML file, refer to XML file contents.
3. Use the **virsh attach-device <vm-name> [--persistent] <xml-filename>** command to hot-plug the interface into the router.

   Note: The **virsh attach-device** command does not check whether a MAC address is already assigned to another network interface. As a result, if you assign the same MAC address to multiple network interfaces and try to configure them, a router might display error messages. To avoid these error messages, ensure that the MAC addresses you assign to hot-plugged interfaces are unique.

Creating XML files for hot-plugging interfaces

On the KVM platform, before hot-plugging an interface into a router, you must create an XML file on the host VM. This XML file describes the parameter of the network interface.

XML file contents

The following table describes the elements that the XML file for a hot-plugged interface can contain.

Table 42. XML file contents

Element	Description
interface type	Interface type. The following values are supported: • network: Specifies a network interface. • bridge: Specifies a bridge interface. Use this value when hot-plugging an interface that is connected to a Spirent port. • direct: Specifies a management interface.
mac address	MAC address of the interface. You must ensure that this address is unique.
source network	(Applies to network interfaces only) Label or name of the network to which the interface connects.
source bridge	(Applies to bridge interfaces only) Label or name of the bridge device to which the interface connects.
model type	Type of the network virtualization model. Currently, the only supported model on the KVM platform is virtio.

Table 42. XML file contents (continued)

Element	Description
address type	(Optional) The virtualized PCI slot into which the interface is plugged. You must use the hexadecimal notation to specify the slot number.

XML file examples

The following table lists a few router hot-plugging scenarios. For each scenario, this table shows the contents of the corresponding sample XML file.

Table 43. Sample hot-plugging XML files

Hot-plugging Scenario	Description

XML file for hot-plugging a network interface and connecting it to the net300 network. A PCI slot is not specified.
Note: The net300 network is created using virt-manager.
	`<interface type='network'>`
	`   <mac address='52:54:00:ff:ff:ff'/>`
	`   <source network='net300'/>`
	`   <model type='virtio'/>`
	` </interface>`

XML file for hot-plugging a network interface into the tenth PCI slot.	`<interface type='network'>`
	`   <mac address='52:54:00:ff:ff:ff'/>`
	`   <address type='pci' domain='0x0000' bus='0x00' slot='0x0a'/>`
	`   <source network='net300'/>`
	`   <model type='virtio'/>`
	`</interface>`

XML file for hot-plugging a network interface and connecting it to a Spirent interface. A PCI slot is not specified.	`<interface type='bridge'>`
	`   <mac address='52:54:00:15:d6:bd'/>`
	`   <source bridge='br2'/>`
	`   <model type='virtio'/>`
	`</interface>`

XML file for hot-plugging a management interface. A PCI slot is not specified.	`<interface type='direct'>`
	`   <mac address='52:54:00:d2:87:1d'/>`
	`   <source dev='em1' mode='bridge'/>`
	`   <model type='virtio'/>`
	`</interface>`

XML file for hot-plugging a management interface. The interface is to be hot-plugged into the third PCI slot.	`<interface type='direct'>`
	`   <mac address='52:54:00:d2:87:1d'/>`
	`   <source dev='em1' mode='bridge'/>`
	`   <model type='virtio'/>`
	`   <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>`
	`</interface>`

Naming sequence

The following table describes how a router assigns names to interfaces and whether they persist.

Table 44. router interface naming on the KVM platform

Hot-plugging sequence example	Naming Sequence	Naming sequence after restarting the router	Naming sequence after shutting down the router and starting it
A nonpersistent interface is hot-plugged.	The interface is hot-plugged into the next available PCI slot	dp0s3	The dp0s3 interface no longer exists.

Table 44. router interface naming on the KVM platform (continued)

Hot-plugging sequence example	Naming Sequence	Naming sequence after restarting the router	Naming sequence after shutting down the router and starting it
	with the higher slot number. For example, dp0s3.		
A persistent interface is hot-plugged.	The interface is hot-plugged into the next available PCI slot with the higher slot number. For example, dp0s3.	dp0s3	The dp0s3 interface persists.
A nonpersistent interface is hot-plugged followed by another nonpersistent interface.	The interfaces are hot-plugged into the next available and consecutive PCI slots with the higher numbers. For example, dp0s3 followed by dp0s4.	dp0s3 dp0s4	The dp0s3 and dp0s4 interfaces no longer exist.
A nonpersistent interface is hot-plugged followed by a persistent interface.	The interfaces are hot-plugged into the next available and consecutive PCI slots with the higher numbers. For example, dp0s3 followed by dp0s4.	dp0s3 dp0s4	The dp0s3 interface is detached and the dp0s4 interface persists, but it is plugged into PCI slot 3 (dp0s3).
A persistent interface is hot-plugged followed by a nonpersistent interface.	The interfaces are hot-plugged into the next available and consecutive PCI slots with the higher numbers. For example, dp0s3 followed by dp0s4.	dp0s3 dp0s4	The dp0s3 interface persists, but the dp0s4 interface is detached and no longer exists.
A persistent interface is hot-plugged into PCI slot 10.	The interface is hot-plugged into PCI slot 10.	dp0s10	The dp0s10 interface persists.
A nonpersistent interface is hot-plugged into PCI slot 10.	The interface is hot-plugged into PCI slot 10.	dp0s10	The dp0s10 interface is detached and no longer exists.
A nonpersistent interface is hot-plugged followed by a persistent interface that is hot-plugged into PCI slot 10.	The nonpersistent interface is plugged into the next available slot with the higher number (for example, dp0s3). The persistent interface takes slot 10.	dp0s3 dp0s10	The dp0s10 interface persists. The dp0s3 interface is detached and no longer exists.
A persistent interface is hot-plugged followed by a nonpersistent interface that is hot-plugged into PCI slot 10.	The persistent interface takes the next available higher slot. The nonpersistent interface takes slot 10.	dp0s3 dp0s10	The dp0s3 interface persists. The dp0s10 interface is detached and no longer exists.
A persistent interface is hot-plugged followed by a persistent interface that is hot-plugged into PCI slot 10.	The first persistent interface takes the next available higher slot. The second persistent interface takes slot 10.	dp0s3 dp0s10	The dp0s3 interface persists. The dp0s10 interface persists.
A nonpersistent interface is hot-plugged followed by a nonpersistent interface that is hot-plugged into PCI slot 10.	The first nonpersistent takes the next available higher slot. The second nonpersistent interface takes slot 10.	dp0s3 dp0s10	The dp0s3 interface is detached and no longer exists. The dp0s10 interface is detached and no longer exists.
A persistent interface is hot-plugged followed by a persistent interface (dp0s4) and then the former interface is detached.	The second interface persists.	dp0s4	The dp0s4 interface persists.

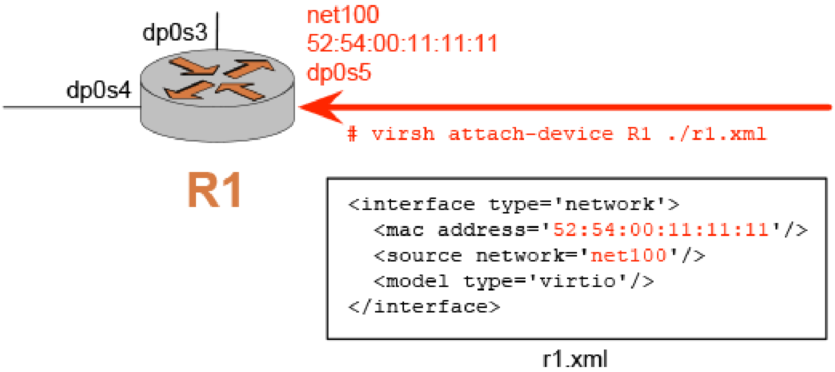
Interface hot-plugging examples (KVM)

This section provides examples of how to hot-plug network interfaces into router that are running in a host KVM system.

Hot-plugging a nonpersistent network interface

The following figure shows how to hot-plug a nonpersistent network interface into a router and connect the interface to the net100 network.

Figure 10. Hot-plugging a nonpersistent interface



```
# virsh attach-device R1 ./r1.xml
```

```
<interface type='network'>
  <mac address='52:54:00:11:11:11'/>
  <source network='net100'/>
  <model type='virtio'/>
</interface>
```

To configure hot-plugging for the scenario that is shown in this figure, perform the following steps on the host VM:

1. Log in to the host VM.
2. Change the directory to /home/vyatta/.
3. Create the r1.xml file and set its contents to the following:

```
<interface type='network'>
   <mac address='52:54:00:11:11:11'/>
   <source network='net100'/>
   <model type='virtio'/>
</interface>
```

4. Save the r1.xml file in the /home/vyatta/ directory.
5. Hot-plug the interface into the R1 router by entering the following command:

```
# virsh attach-device R1 ./r1.xml
```

The router hot-plugs the interface into the next available PCI slot with the higher number.

Note: In this instance, which is the default case, the interface is nonpersistent. This means that the interface is automatically detached during the shutdown sequence of the guest VM.

To detach this interface, enter the following command:

```
# virsh detach-device R1 ./r1.xml
```

Hot-plugging a persistent network interface

To hot-plug the interface that is specified by the r1.xml file (shown in the figure in Hot-plugging a nonpersistent network interface) in a persistent manner, enter the following command:

```
# virsh attach-device R1 --persistent ./r1.xml
```

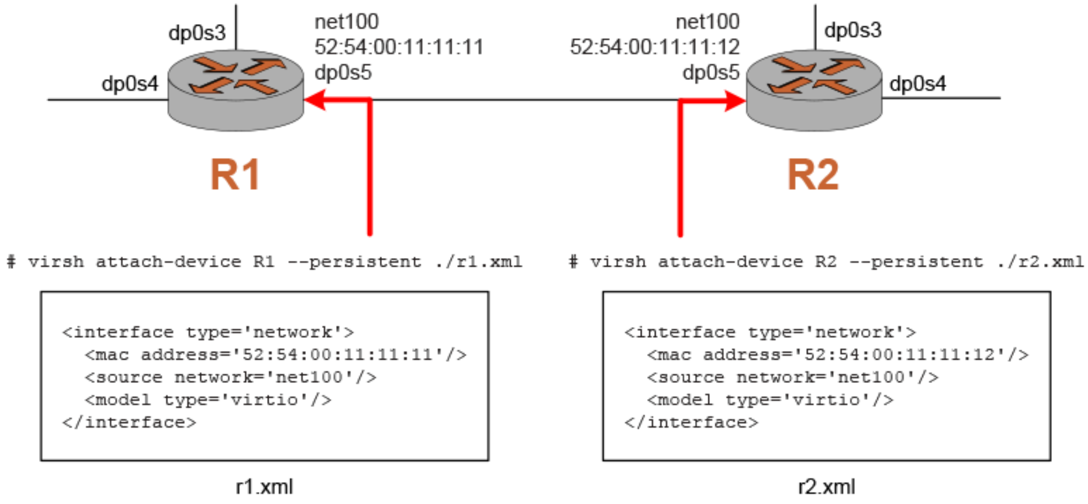
For a persistent interface, to detach the interface in a persistent manner, enter the following command on your VM guest:

```
# virsh detach-device R1 --persistent ./r1.xml
```

Hot-plugging two persistent network interfaces to connect two routers

The following figure shows how to hot-plug two persistent network interfaces on two routers so that the two routers are connected to the same network.

Figure 11. Connecting two routers with hot-plugged interfaces



```
# virsh attach-device R1 --persistent ./r1.xml
```

```
<interface type='network'>
  <mac address='52:54:00:11:11:11'/>
  <source network='net100'/>
  <model type='virtio'/>
</interface>
```

```
# virsh attach-device R2 --persistent ./r2.xml
```

```
<interface type='network'>
  <mac address='52:54:00:11:11:12'/>
  <source network='net100'/>
  <model type='virtio'/>
</interface>
```

To configure hot-plugging for the scenario that is shown in this figure, perform the following steps on the host VM:

1. Log in to the VM guest.
2. Change the directory to /home/vyatta/.
3. Create the r1.xml file and set its contents to the following:

```
<interface type='network'>
   <mac address='52:54:00:11:11:11'/>
   <source network='net100'/>
   <model type='virtio'/>
</interface>
```

4. Save the r1.xml file in the /home/vyatta/ directory.
5. Create the r2.xml file and set its contents to the following:

```
<interface type='network'>
   <mac address='52:54:00:11:11:12'/>
   <source network='net100'/>
   <model type='virtio'/>
</interface>
```

   The only difference between r1.xml and r2.xml is the MAC address. Because these two interfaces are on the same network, the MAC addresses must be unique.
6. Save the r2.xml file in the /home/vyatta/ directory.
7. Hot-plug an interface into the R1 router by entering the following command:

```
# virsh attach-device R1 --persistent./r1.xml
```

8. Hot-plug an interface into the R2 router by entering the following command:

```
# virsh attach-device R2 --persistent./r2.xml
```

Commands for attaching and detaching interfaces on the KVM platform

You can use the following KVM commands to attach and detach network interfaces from a router. These commands are available through the libvirt library on the KVM platform.

- virsh attach-device
- virsh detach-device

virsh attach-device

Attaches a network interface to a router.

virsh attach-device *vm-name* [**--persistent**] *xml-filename*

Nonpersistent. The interface is detached during the router shutdown sequence.

vm-name

The name of the guest VM (the router).

persistent

Causes the interface to remain attached to the device after the router is powered on after being shut down.

xml-filename

The name of the XML file that specifies the interface parameters.

Use this command to attach a network interface to a router. If you do not use the **persistent** keyword, the interface is detached during the router shutdown sequence. To ensure that the interface remains attached, use the **persistent** keyword.

virsh detach-device

Detaches a network interface from a router.

```
virsh detach-device  vm-name [ --persistent ] xml-filename
```

Nonpersistent. The interface is reattached after VM restarts or is powered on.

vm-name

The name of the guest VM (the router).

persistent

Causes the interface to remain detached from the device after the router is powered on after being shut down.

xml-filename

The name of the XML file that specifies the interface parameters.

Use this command to detach a network interface from a router. If you do not use the **persistent** keyword, the interface remains attached, even after the router is restarted or powered on after being shut down. To ensure that the interface remains detached, use the **persistent** keyword.

If you hot-plug two interfaces and give them the same MAC address by mistake, use this command to detach one of these interfaces. In the XML file, specify the PCI slot to which the interface is plugged, as shown in the following example:

```
<interface type='network'>
<mac address='52:54:00:dd:dd:dd'/>
<address type='pci' domain='0x0000' bus='0x00' slot='0x08' function='0x0'/>
<source network='net200'/>
<model type='virtio'/>
</interface>
```

Chapter 18. Logging

This chapter describes the router logging mechanism.

Logging configuration

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, 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 include notices of ordinary and routine operations as well as warnings, failure, and error messages.

The logging function of the router uses the UNIX `syslogd` process. Logging configuration performed within the system CLI is stored in the `/etc/sylogd.conf` file.

By default, local logging is enabled and sends messages to the `/var/log/messages` file.

Logging facilities

The router supports the following standard syslog facilities.

Table 45. Syslog facilities

Facility	Description
auth	Authentication and authorization
authpriv	Nonsystem authorization
cron	Cron daemon
daemon	System daemons
kern	Kernel
lpr	Line printer spooler
mail	Mail subsystem
mark	Time stamp
news	USENET subsystem
security	Security subsystem
syslog	System logging
user	Application processes
uucp	UUCP subsystem

Table 45. Syslog facilities (continued)

Facility	Description
local0	Local facility 0 (unused)
local1	Local facility 1 (unused)
local2	Local facility 2 (unused)
local3	Local facility 3 (unused)
local4	Local facility 4 (unused)
local5	Local facility 5 (user commands)
local6	Local facility 6 (data plane)
local7	Local facility 7 (routing protocols)
all	All facilities excluding “mark”

In addition, logging can be selectively enabled for some specific routing components. For more information, refer to Enabling and disabling logging for specific features.

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, a named file in the local file system, a server running the `syslogd` utility (that is, a syslog server), or 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, multiple users, or all users logged in to 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 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 with log files. When log messages are written to a file, the system writes up to 250 KB of log messages into the *logfile* file, where *logfile* is either the main log file or a name you have assigned to a user-defined file. When the log file reaches its maximum size, the system closes it and compresses it into an archive file. The archive file is named *logfile*.1.gz.

At this point, the logging utility opens a new log file and begins to write system messages to it. When the new log file is full, the first archive file is renamed *logfile*.2.gz and the new archive file is named *logfile*.1.gz.

The system archives log files in this way until a maximum number of log files exist. By default, the maximum number of archive files is five (that is, up to *logfile*.5.gz), where *logfile*.1.gz always represents the most recent file. After the fifth file, the oldest archive log 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 with the following parameters.

- Use the size parameter to specify the maximum size of each archive log file.
- Use the files parameter to specify the maximum number of archive files to be maintained.

Log severities

Log messages generated by the router are associated with one of the following levels of severity.

Table 46. 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.
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	Debugging level. Trace-level information is being provided.

CAUTION:

Risk of service degradation. Debugging severity is resource intensive. Setting logging levels to Debug can affect performance.

Logging configuration example

Table 47: Creating a log file to capture kernel-related alerts of critical and higher severity shows how to create 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.

Table 47. Creating a log file to capture kernel-related alerts of critical and higher severity

Step	Command

Create a log file 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#`

Verify the configuration.	`vyatta@R1# show system syslog file kernel-log`
	
	` facility kern {`
	`    level crit`
	` }`

The `show log file` *kernel-log* command can then be used in operational mode to display the contents of the *kernel-log* log file.

Enabling and disabling logging for specific features

Some features of the 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 the `/var/log/user` directory.

Overriding the host syslog logging facility for log entries

The system syslog configuration on a router provides the capability to send log entries to a host system. You can designate which log entries are sent by specifying the facility and log level by using the system syslog global facility level command. You can use this command multiple times to specify different facility values for the log entries sent to the specified host.

You can also configure a facility override value that replaces the facility fields in all log entries sent to a specified host. For example, you can specify multiple facility values for a set of log entries sent by the router to a log host. Before sending the entries to the host, the facility values are replaced with the override value. When the host receives these log entries, their facility field value is designated by the facility override value.

Note:

A facility override only affects log entries sent to a host and does not affect entries set to a user, console, or file.

A facility override is specific to a host. Different hosts can have different override values.

The following example provides a configuration of multiple host syslog logging facilities with a facility overrride.

Step	Command
Specify the facilities messages that are sent to a host.	vyatta@R1# set system syslog host 10.10.10.10 facility auth level crit vyatta@R1# set system syslog host 10.10.10.10 facility user level err
Specify the facility override value that replaces the facility values of the log entries sent to the host.	vyatta@R1# set system syslog host 10.10.10.10 facility-override local7
Commit the configuration.	vyatta@R1# commit

Verify the configuration.	vyatta@R1#show system syslog
	syslog {
	    host 10.10.10.10 {
	        facility auth {
	            level crit
	        }
	        facility user {
	            level err
	        }
	        facility-override local7
	    }
	}

Configuring rate limiting for syslog logging

You can use the following commands to enable and configure rate limiting for syslog logging. These commands let you set the burst count, which specifies the maximum number of messages that a process can log during a logging interval, and the logging interval.

- system syslog rate-limit burst

  Limits the number of messages that a process can log during a logging interval
- system syslog rate-limit interval

  Sets the logging interval for processes

Enabling and disabling rate limiting

To enable rate limiting for syslog logging, you must configure the burst count, logging interval, or both.

If you configure only the burst count, the logging interval is 5 seconds (default). For example, if you set the burst count to 300, a process can log up to 300 messages every 5 seconds. The system discards extra messages and adds a syslog entry to that effect.

If you configure only the logging interval, the burst count is 200 (default). For example, if you set the interval to 20 seconds, a process can log up to 200 messages every 20 seconds. The system discards extra messages and adds a syslog entry to that effect.

If you configure the logging interval and burst count, the system enforces the configured limits. For example, if you set the burst count to 250 and the logging interval to 10 seconds, a process can log up to 250 messages every 10 seconds. The system discards extra messages and adds a syslog entry to that effect.

The system enforces syslog rate limiting on a single-process basis. For example, if the burst count is 200 and the logging interval is 10 seconds, each process running on the system can log up to 200 messages during the 10-second interval.

To disable rate limiting, use the **delete system syslog rate-limit** command. When rate limiting is disabled, there are no limits on the number of messages that a process can log.

Configuring the burst count

To limit the number of messages that a process can log during an interval to 250, perform the following steps in configuration mode.

Step	Command
Set the burst count.	`vyatta@vyatta# set system syslog rate-limit burst 250`
Commit the configuration.	`vyatta@vyatta# commit`

Verify the configuration.	`vyatta@vyatta# show system syslog rate-limit`
	` rate-limit {`
	`        burst 250`
	` }`

Configuring the logging interval

To set the logging interval to 10 seconds, perform the following steps in configuration mode.

Step	Command
Set the logging interval.	`vyatta@vyatta# set system syslog rate-limit interval 10`
Commit the configuration.	`vyatta@vyatta# commit`

Verify the configuration.	`vyatta@vyatta# show system syslog rate-limit`
	` rate-limit {`
	`        interval 10`
	` }`

Configuring the burst count and logging interval

To set the burst count to 300 messages and logging interval to 15 seconds, perform the following steps in configuration mode.

Step	Command
Set the burst count.	`vyatta@vyatta# set system syslog rate-limit burst 300`

Step	Command
Set the logging interval.	`vyatta@vyatta# set system syslog rate-limit interval 15`
Commit the configuration.	`vyatta@vyatta# commit`

Verify the configuration.	`vyatta@vyatta# show system syslog rate-limit`
	` rate-limit {`
	`        burst 300`
	`        interval 15`
	` }`

Logging all user commands

You can configure the syslog system to log all commands that a user runs to a local or remote syslog destination.

The syslog system uses specific values of the following attributes to identify user-command log entries:

- Facility: Log entries for user commands have a facility value of local5.
- Severity level: Log entries for user commands have a severity level of info.

To log user commands, configure the syslog system to send log entries with a local5 facility and info severity level to a local or remote syslog destination. A local destination is a user-defined file. A remote destination is a syslog server.

The following example shows how to configure the syslog system to log user commands to the /var/log/user/cmds.log file.

Step	Command
Configure the system to send user-command log entries to the cmds.log file. If not present, this command creates the cmds.log file in the /var/log/user directory. **Note:** For security reasons, the system restricts user-defined files to the /var/log/user directory, which is why this commands requires only the filename (in this example, cmds.log) and not the path. If the filename includes the path, this command returns an error.	`vyatta@vyatta# set system syslog file cmds.log facility local5 level info`
Commit the configuration.	`vyatta@vyatta# commit`

Verify the configuration.	`vyatta@vyatta# show system syslog file`
	` file cmds.log {`
	`        facility local5 {`
	`                level info`
	`        }`
	` }`

The following example shows entries from the cmds.log file.

```
2017-01-17T17:13:19.876844+00:00 localhost -vbash[3392]: HISTORY: PID=3392
 UID=1000 configure
```

```
2017-01-17T17:15:06.544493+00:00 localhost vbash[3641]: HISTORY: PID=3641
 UID=1000 set protocols rip interface lo
2017-01-17T17:16:10.351281+00:00 localhost vbash[3641]: HISTORY: PID=3641
 UID=1000 set protocols ospf
2017-01-17T17:16:33.016625+00:00 localhost vbash[3641]: HISTORY: PID=3641
 UID=1000 set protocols ospf log lsa all
2017-01-17T17:17:11.450432+00:00 localhost vbash[3641]: HISTORY: PID=3641
 UID=1000 commit
```

The following example shows how to configure the syslog system to log user commands to the syslog server at 192.168.1.2.

Step	Command
Configure the system to send user-command log entries to the syslog server at 192.168.1.2.	`vyatta@vyatta# set system syslog host 192.168.1.2 facility local5 level info`
Commit the configuration.	`vyatta@vyatta# commit`

Verify the configuration.	`vyatta@vyatta# show system syslog host`
	`host 192.168.1.2 {`
	`        facility local5 {`
	`                level info`
	`        }`
	`}`

Chapter 19. Logging Commands

delete log file

Deletes a user-defined log file, including all its archive files.

```
delete log file  file-name
```

file-name

A user-defined log file in the **/var/log/user** directory.

Operational mode

Use this command to delete a user-defined log file.

User-defined log files are created in the `/var/log/user` directory. When you enter this command, the specified file and all associated archive files are deleted from this directory.

Note that deleting the user-defined log file does not stop the system from logging events. If you use this command while the system is logging events, old log events are deleted, but events after the delete operation are recorded in the new file. To delete the file altogether, first disable logging to the file by using the system syslog, and then delete it.

show log

Displays the contents of a log file or files.

```
show log  [ all | authorization | directory | file file-name | tail [ lines ]
| component ]
```

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

The number of lines that **tail** displays at the end of the system log.

component

A specific system component. The component is any of the following:

- dhcp—Displays the log for dhcp
- dns—Displays the log for dns
- firewall—Displays the log for firewall
- https—Displays the log for https
- image—Displays the log from an image
- nat—Displays the log for nat
- openvpn—Displays the log for openvpn
- snmp—Displays the log for snmp
- vpn—Displays the log for vpn
- vrrp—Displays the log for vrrp

Operational mode

Use this command to display 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 the main system log file. User-specified log files are defined by using the system syslog file facility level.

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 continues to display log messages as they are added to the log file. This command can be interrupted by using <Ctrl+C>.

When *lines* is specified, the last *lines* lines of the system log are to be displayed.

When *component* is specified, log messages that relate to that component are displayed.

show log image

Displays the contents of a log file or files on an image other than the currently active image.

```
show log image  image-name [ all | authorization | directory | file file-name |
tail [ lines ] ]
```

When used with no option, this command displays the contents of the main system log. The system writes syslog messages to this default log.

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 10 lines of the system log for the specified image.

lines

The number of lines to be displayed. If not specified, 10 lines are displayed.

Operational mode

Use this command to display the contents of a log file or files on an image other than the currently active image.

system syslog

Configures the syslog utility of the system.

```
set system syslog
```

```
delete system syslog
```

```
show system syslog
```

Configuration mode

```
system {
    syslog {
    }
}
```

Use this command to configure the syslog utility of the system.

Using this command, you can set the destinations for log messages from different routing components (facilities) and specify what severity level of message should be reported for each facility.

Log messages generated by the router are associated with one of the following levels of severity.

Table 48. 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.

Table 48. Syslog message severities (continued)

Severity	Meaning
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	Debugging level. Trace-level information is being provided.

The router supports the following standard syslog facilities.

Table 49. Syslog facilities

Facility	Description
auth	Authentication and authorization
authpriv	Nonsystem authorization
cron	Cron daemon
daemon	System daemons
kern	Kernel
lpr	Line printer spooler
mail	Mail subsystem
mark	Time stamp
news	USENET subsystem
security	Security subsystem
syslog	System logging
user	Application processes
uucp	UUCP subsystem
local0	Local facility 0 (unused)
local1	Local facility 1 (unused)
local2	Local facility 2 (unused)
local3	Local facility 3 (unused)
local4	Local facility 4 (unused)
local5	Local facility 5 (user commands)
local6	Local facility 6 (data plane)
local7	Local facility 7 (routing protocols)
all	All facilities excluding “mark”

Messages are written either to the main log file (the default) or 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 with log files. When log messages are written to a file, the system writes up to 250 KB of log messages into the *logfile* file, where *logfile* is either the main log file or a name you have assigned to a user-defined file. When the log file reaches its maximum size, the system closes it and compresses it into an archive file. The archive file is named *logfile*.1.gz.

At this point, the logging utility opens a new log file and begins to write system messages to it. When the new log file is full, the first archive file is renamed *logfile*.2.gz and the new archive file is named *logfile*.1.gz.

The system archives log files in this way until a maximum number of log files exist. By default, the maximum number of archive files is 5 (that is, up to *logfile*.5.gz), where *logfile*.1.gz always represents the most recent file. After the fifth file, the oldest archive log 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 with the following parameters.

- Use the size parameter to specify the maximum size of each archive 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 level

Specifies which messages are sent to the console.

```
set system syslog console facility  facility level level
```

```
delete system syslog console facility  [ facility [ level ] ]
```

```
show system syslog console facility  [ facility [ level ] ]
```

facility

Multi-node. The kinds of messages that are sent to the console. Refer to “Usage Guidelines” for the 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 level of log message that are reported to the console. The level is any of `emerg`, `alert`, `crit`, `err`, `warning`, `notice`, `info`, or `debug`. Refer to “Usage Guidelines” for the system syslog command for the meanings of these levels.

By default, messages of `err` severity are logged to the console.

Configuration mode

```
system {
    syslog {
        console {
            facility facility {
                level level
            }
        }
    }
}
```

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 display the configuration of console messages.

system syslog file archive

Specifies the settings for log file archiving of a user-defined log file.

`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** }

filename

Multi-node. A file to which the specified log messages are written. A file name 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

The maximum number of archive files that are maintained for this log file. After the maximum number has been reached, logs are rotated with the oldest file being overwritten. The default maximum number is 10.

size

The maximum size in bytes of archive files for this log file. After the maximum has been reached, the file is closed and archived in compressed format. The default maximum size is 1 MB.

Configuration mode

```
system {
    syslog {
        file filename{
            archive {
                files files
                size size
            }
        }
    }
}
```

Use this command to specify settings for log file archiving of a user-defined log file.

Use the `set` form of this command to specify settings for log file archiving of a user-defined log file.

Use the `delete` form of this command to restore the default archiving configuration for a user-defined log file.

Use the `show` form of this command to display the configuration of the user-defined log file archiving.

system syslog file facility level

Specifies which messages are sent to a user-defined log file.

```
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 ] ]
```

filename

Multi-node. A file to which the specified log messages are written. A file name 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 are 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 level of log message that are reported. The level is any of `emerg`, `alert`, `crit`, `err`, `warning`, `notice`, `info`, or `debug`. Refer to “Usage Guidelines” for the system syslog command for the meanings of these levels.

By default, messages of `warning` severity are logged to the file.

The router supports the sending of log messages to the main system log file, the console, a remote host, a user-specified file, or a user account.

Configuration mode

```
system {
    syslog {
        file filename {
            facility facility {
                level level
            }
        }
    }
}
```

Use this command to specify which messages are sent to a user-defined log file.

Use the `set` form of this command to specify which messages are sent to a user-defined log file.

Use the `delete` form of this command to restore the default message configuration for a user-defined log file.

Use the `show` form of this command to display the configuration for user-defined log file messages.

system syslog global archive

Specifies the settings for log file archiving of the main system log file.

`set system syslog global archive` { **files** *files* | **size** *size* }

`delete system syslog global archive` { **files** | **size** }

`show system syslog global archive` { **files** | **size** }

files

The maximum number of archive files that are maintained for the main system log file. After the maximum has been reached, logs are rotated with the oldest file being overwritten. The default maximum number is 10.

size

The maximum size in bytes of archive files for the main system log file. After the maximum has been reached, the file is closed and archived in compressed format. The default maximum size is 1 MB.

Configuration mode

```
system {
    syslog {
        global {
            archive {
                files files
                size size
            }
        }
    }
}
```

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 configuration for log file archiving.

Use the `show` form of this command to display the configuration for log file archiving.

system syslog global facility level

Specifies which messages are sent to the main system log file.

`set system syslog global facility` *facility* **level** *level*

`delete system syslog global facility` [*facility* [**level**]]

`show system syslog global facility` [*facility* [**level**]]

facility

Multi-node. The kinds of messages that are sent to the main system log file. Refer to “Usage Guidelines” for the 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 level of log message that are reported. The level is any of `emerg`, `alert`, `crit`, `err`, `warning`, `notice`, `info`, or `debug`. Refer to “Usage Guidelines” for the system syslog command for the meanings of these levels.

By default, messages of `warning` severity are logged to the main system log file.

Configuration mode

```
system {
    syslog {
        global {
            facility facility {
                level level
                }
            }
        }
    }
```

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 configuration for the main system log file.

Use the `show` form of this command to display the configuration for the main system log file.

system syslog host facility level

Specifies which messages are sent to the remote syslog server.

```
set routing routing-instance instance-namesystem syslog host  hostname facility
facility level level

delete routing routing-instance instance-namesystem syslog host  hostname facility
[ facility [ level ] ]

show routing routing-instance instance-namesystem syslog host  hostname facility [
facility [ level ] ]
```

instance-name

A VRF routing instance. Alphanumeric string. The leading character cannot be _ or -. You cannot use default as an instance name; it is reserved for the default routing instance.

hostname

Leaf-list. An IP address or a host name. The host must be running the syslog protocol. A host name can include numbers, letters, hyphens (-), and such other commonly used characters. The IP address must follow one of the addressing standards: *X.X.X.X* or [*x*:*x*:*x*:*x*:*x*:*x*:*x*:*x*]. All host formats may have a :port suffix. The IPv6 address must be enclosed in square brackets ([ ]) to delimit the address and port.

You can send log messages to multiple hosts by creating multiple `host` configuration nodes.

facility

Leaf-list. The kinds of messages that are sent to the host. Refer to “Usage Guidelines” for the `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 level of log message that are reported. The level is any of `emerg`, `alert`, `crit`, `err`, `warning`, `notice`, `info`, or `debug`. Refer to “Usage Guidelines” for the system syslog command for the meanings of these levels.

By default, messages of `err` severity are logged to hosts.

Configuration mode

```
routing {
   routing-instance {
             system {
                syslog {
                    host {
                        facility {
                            level
                         }
                     }
                 }
               }
          }
}
```

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 file message configuration for the remote syslog server log.

Use the **show** form of this command to display the configuration for the remote syslog server.

system syslog host facility-override

Defines the facility override value that replaces the facility fields for all log entries sent to the specified host.

`set system syslog host` *hostname* **facility-override** *facility*

`delete system syslog host` *hostname* **facility-override**

`show system syslog` [**host** *hostname* **facility-override** [*facility*]]

hostname

An IP address or a host name configured with system syslog host facility level.

facility

The overrride facility value that replaces the facility fields for all log entries to the specified host messages. “Usage Guidelines” for the `system syslog` command lists the supported logging facilities.

Configuration mode

```
system {
    syslog {
        host hostname {
            facility-override facility
        }
    }
}
```

A facility override only affects log entries sent to a host and does not affect entries set to a user, console, or file.

A facility override is specific to a host. Different hosts can have different override values.

Use the `set` form of this command to define the facility override value for all log entries sent to the specified host.

Use the `delete` form of this command to restore the default facility values for the log entries sent to the specified host.

Use the **show** form of this command to display the configuration for the facility override configuation.

system syslog rate-limit burst

Sets the burst count, that is, the maximum number of messages that a process can add to syslog during a logging interval.

```
set system syslog rate-limit burst   number-of-log-entries
```

```
delete system syslog rate-limit burst
```

```
show system syslog rate-limit burst
```

number-of-log-entries

The burst count, a number that ranges from 1 through 4294967295 (4,294,967,295).

Configuration mode

```

system {
    syslog {
        rate-limit{
            burst number-of-log-entries
        }
    }
}
```

Use this command to set the burst count. The system discards the extra messages that exceed the burst count and adds a syslog entry to that effect. To set the logging interval, refer to system syslog rate-limit interval. For more information about rate limiting for syslog logging, refer to Configuring rate limiting for syslog logging.

Use the `set` form of this command to set the burst count.

Use the `delete` form of this command to reset the burst count to 200 messages (default).

Note: If the interval for syslog logging is not configured, using the `delete` form of this command disables syslog rate liming; there are no limits on how many messages a process can log.

Use the `show` form of this command to display the burst count.

system syslog rate-limit interval

Sets the interval (in seconds) during which the system allows a process to log as many as the maximum number of messages that is specified by the burst count parameter for syslog logging.

```
set system syslog rate-limit interval  logging-interval
```

```
delete system syslog rate-limit interval
```

```
show system syslog rate-limit interval
```

logging-interval

The interval (in seconds) during which a process can log as many as the maximum number of messages that is specified by the burst count parameter for syslog logging. The interval ranges from 1 through 4294967295 seconds (4,294,967,295).

Configuration mode

```
system {
    syslog {
        rate-limit{
            interval logging-interval
        }
    }
}
```

Use this command to set the syslog logging interval. To set the maximum number of messages that a process can log during a logging interval, refer to system syslog rate-limit interval. For more information about rate limiting for syslog logging, refer to Configuring rate limiting for syslog logging.

Use the `set` form of this command to set the logging interval.

Use the `delete` form of this command to reset the logging interval to 5 seconds (default).

Note: If the burst count for syslog logging is not configured, using the `delete` form of this command disables syslog rate liming; there are no limits on how many messages a process can log.

Use the `show` form of this command to display the configured logging interval.

system syslog user facility level

Specifies which messages are sent to the terminal of a user.

```
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 ] ]
```

userid

Multi-node. A user ID.

You can send log messages to multiple users by creating multiple `user` configuration nodes.

facility

Multi-node. The kinds of messages that are sent to the user. Refer to “Usage Guidelines” for the 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 level of log message that are reported to the user. The level is any of `emerg`, `alert`, `crit`, `err`, `warning`, `notice`, `info`, or `debug`. Refer to “Usage Guidelines” for the system syslog command for the meanings of these levels.

By default, messages of `err` severity are logged to the terminal of the user.

Configuration mode

```
system {
    syslog {
        user userid {
            facility facility {
                level level
            }
        }
    }
}
```

Use this command to specify which messages are sent to the terminal of a user.

Use the `set` form of this command to specify which messages are sent to the terminal of a user.

Use the `delete` form of this command to restore the default configuration of terminal messages for a user.

Use the `show` form of this command to display the configuration of terminal messages for a user.

Chapter 20. Loopback and Data Plane Interfaces

Following are the supported formats of the interface name:

- `lo` or `lo`*n*The name of a loopback interface, where *n* ranges from 1 through 99999.
- `dp`*x*`p`*y*`p`*z*—The name of a data plane interface, where
  - `dp`*x* specifies the data plane identifier (ID). Currently, only dp0 is supported.
  - p*y* specifies a physical or virtual PCI slot index (for example, p129).
  - `p`*z* specifies a port index (for example, p1). For example, dp0p1p2, dp0p160p1, and dp0p192p1.
- `dp`*x*`em`*y*—The name of a data plane interface on a LAN-on-motherboard (LOM) device that does not have a PCI slot, where `em`*y* specifies an embedded network interface number (typically, a small number). For example, dp0em3.
- `dp`*x*`s`*y*—The name of a data plane interface in a system in which the BIOS identifies the network interface card to reside in a particular physical or virtual slot *y*, where *y* is typically a small number. For example, for the dp0s2 interface, the BIOS identifies slot 2 in the system to contain this interface.
- `dp`*x*`P`*n*`p`*y*`p`*z*—The name of a data plane interface on a device that is installed on a secondary PCI bus, where `P`*n* specifies the bus number. You can use this format to name data plane interfaces on large physical devices with multiple PCI buses. For these devices, it is possible to have network interface cards installed on different buses with these cards having the same slot ID. The value of *n* must be an integer greater than 0. For example, dp0P1p162p1 and dp0P2p162p1.

Chapter 21. VRF support

VRF support for RADIUS authentication

RADIUS must run on a single routing instance. If you configure a RADIUS server without specifying the routing instance, the RADIUS server starts in the default routing instance. If you specify a nondefault routing instance, you must verify that all servers configured for AAA with the RADIUS server are accessible by way of the same routing instance.

The following examples show excerpts of RADIUS configurations that use these values:

- routing instance = BLUE
- radius-server-address = 42.42.42.42
- secret-code = secured
- port-no = 1820
- timeout = 2

The following example shows how to configure RADIUS for the default routing instance.

```
vyatta@R1# set system login radius-server 42.42.42.42
vyatta@R1# set system login radius-server 42.42.42.42 secret secured
vyatta@R1# set system login radius-server 42.42.42.42 port 1820
vyatta@R1# set system login radius-server 42.42.42.42 timeout 2
vyatta@R1# commit
vyatta@R1# run show configuration
system {
        login {
                radius-server 42.42.42.42 {
                        secret secured
                        port 1820
                        timeout 2
                }
        }
}
```

The following example shows the same configuration sequence for the BLUE routing instance.

```
vyatta@R1# set routing routing-instance BLUE system login radius-server
 42.42.42.42
vyatta@R1# set routing routing-instance BLUE system login radius-server
 42.42.42.42 secret secured
vyatta@R1# set routing routing-instance BLUE system login radius-server
 42.42.42.42 port 1820
vyatta@R1# set routing routing-instance BLUE system login radius-server
 42.42.42.42 timeout 2
vyatta@R1# commit
vyatta@R1# run show configuration
vyatta@R1# routing {
        routing-instance BLUE {
                system {
                        login {
                                radius-server 42.42.42.42 {
                                        secret secured
                                        port 1820
                                        timeout 2
                                }
                        }
                }
        }
}
```

For more information about RADIUS and configuring RADIUS, see *Basic System Configuration Guide*.

VRF support for file transfer client connections

The router uses FTP that contains several commands. If the network configuration supports VRF, the syntax for each command includes optional VRF parameters. The optional VRF parameters specify the non-default VRF that is used when running the command.

FTPs used in commands that support non-default VRFs must access servers on non-default VRFs. Therefore, commands that support non-default VRF also must also be aware of the VRF parameter that is used in the configuration.

For example, you might have router images stored on a server in the non-default VRF; so, the `add system image` command must be able to download from that server. The `add system image` command sytax consists of the routing instance parameter that specifies the non-default VRF that is used. The command follows:

```
vyatta@R1# add system image { iso-filename | [routing-instance <ri-name>]
 iso-URL [ username username password password ] }
```

An example of a routing instance follows:

```
vyatta@R1# add system image routing-instance red
 http://1.2.3.4/images/vrouter.iso
```

Command support for VRF routing instances

VRF allows a router to support multiple routing tables, one for each VRF routing instance. Some commands in this guide support VRF and can be applied to particular routing instances.

Use the guidelines in this section to determine correct syntax when adding VRF routing instances to commands. For more information about VRF, refer to *Basic Routing Configuration Guide*. This guide includes an overview of VRF, VRF configuration examples, information about VRF-specific features, and a list of commands that support VRF routing instances.

Adding a VRF routing instance to a Configuration mode command

For most Configuration mode commands, specify the VRF routing instance at the beginning of a command. Add the appropriate VRF keywords and variable to follow the initial action (**set**, **show**, or **delete**) and before the other keywords and variables in the command.

Configuration mode example: syslog

The following command configures the syslog logging level for the specified syslog host. The command does not include a VRF routing instance, so the command applies to the default routing instance.

```
vyatta@R1# set system syslog host 10.10.10.1 facility all level debug
vyatta@R1# show system syslog
syslog {
    host 10.10.10.1 {
            facility all {
                    level debug
            }
    }
}
```

The following example shows the same command with the VRF routing instance (GREEN) added. Notice that **routing routing-instance GREEN** has been inserted between the basic action (**set** in the example) and the rest of the command. Most Configuration mode commands follow this convention.

```

vyatta@R1# set routing routing-instance GREEN system syslog host 10.10.10.1
 facility all level debug
vyatta@R1# show routing
routing {
     routing-instance GREEN {
             system {
                     syslog {
                            host 11.12.13.2:514 {
                                     facility all {
                                             level debug
                                     }
                             }
                     }
             }
     }
}
```

Configuration mode example: SNMP

Some features, such as SNMP, are not available on a per-routing instance basis but can be bound to a specific routing instance. For these features, the command syntax is an exception to the convention of specifying the routing instance at the beginning of Configuration mode commands.

The following example shows how to configure the SNMPv1 or SNMPv2c community and context for the RED and BLUE routing instances. The first two commands specify the RED routing instance as the context for community A and BLUE routing instance as the context for community B. The subsequent commands complete the configuration.

For more information about configuring SNMP, refer to *Remote Management Configuration Guide*.

```
vyatta@R1# set service snmp community commA context RED
vyatta@R1# set service snmp community commB context BLUE
vyatta@R1# set service snmp view all oid 1
vyatta@R1# set service snmp community commA view all
vyatta@R1# set service snmp community commB view all
vyatta@R1# show service snmp community
 community commA {
        context RED
        view all
 }
 community commB {
        context BLUE
        view all
 }
[edit]
vyatta@vyatta#
```

Adding a VRF routing instance to an Operational mode command

The syntax for adding a VRF routing instance to an Operational mode command varies according to the type of command parameters:

- If the command does not have optional parameters, specify the routing instance at the end of the command.
- If the command has optional parameters, specify the routing instance after the required parameters and before the optional parameters.

Operational mode examples without optional parameters

The following command displays dynamic DNS information for the default routing instance.

```
vyatta@vyatta:~$ show dns dynamic status
```

The following command displays the same information for the specified routing instance (GREEN). The command does not have any optional parameters, so the routing instance is specified at the end of the command.

```
vyatta@vyatta:~$ show dns dynamic status routing-instance GREEN
```

Operational mode example with optional parameters

The following command obtains multicast path information for the specified host (10.33.2.5). A routing instance is not specified, so the command applies to the default routing instance.

```
vyatta@vyatta:~$ mtrace 10.33.2.5 detail
```

The following command obtains multicast path information for the specified host (10.33.2.5) and routing instance (GREEN). Notice that the routing instance is specified before the optional **detail** keyword.

```
vyatta@vyatta:~$ mtrace 10.33.2.5 routing-instance GREEN detail
```

Operational mode example output: SNMP

The following SNMP **show** commands display output for routing instances.

```
vyatta@vyatta:~$ show snmp routing-instance
Routing Instance SNMP Agent is Listening on for Incoming Requests:
Routing-Instance            RDID
-----------------           ----
RED                         5

vyatta@vyatta:~$ show snmp community-mapping
SNMPv1/v2c Community/Context Mapping:
Community                   Context
---------                   -------
commA                       'RED'
commB                       'BLUE'
deva                        'default'


vyatta@vyatta:~$ show snmp trap-target
SNMPv1/v2c Trap-targets:
Trap-target                   Port   Routing-Instance Community
-----------                   ----   ---------------- ---------
1.1.1.1                              'RED'           'test'


vyatta@vyatta:~$ show snmp v3 trap-target
SNMPv3 Trap-targets:
```

```
Trap-target                   Port   Protocol Auth Priv Type   EngineID
          Routing-Instance User
-----------                   ----   -------- ---- ---- ----   --------
          ---------------- ----
2.2.2.2                       '162'  'udp'    'md5      'infor
          'BLUE'           'test'
```

Chapter 22. Baseboard Management Controller (BMC) Commands

set system bmc health-check

Enables baseboard management controller (BMC) monitoring.

```
set system bmc health-check
```

```
delete system bmc health-check
```

```
show system bmc health-check
```

Configuration mode.

```
system {
    bmc {
        health-check
    }
}
```

Use the `set` form of this command to enable BMC monitoring.

Use the `delete` form of this command to disable BMC monitoring.

Use the `show` form of this command to display BMC monitoring.

set system bmc sel

Configures the baseboard management controller (BMC) system event log (SEL).

```
set system bmc sel {syslog|{mode {circular | capacity }}}
```

```
delete system bmc sel {syslog|{mode {circular | capacity }}}
```

```
show system bmc sel {syslog|{mode {circular | capacity }}}
```

syslog

Log system events to syslog.

circular

When the system event log is full, add new events, deleting the oldest events.

capacity

When the system event log is full, do not add new events .

Configuration mode

```
system {
    bmc {
        sel
            {
                mode
                syslog
             }
    }
}
```

Use the `set` form of this command to configure the BMC system event log.

Use the `delete` form of this command to delete the BMC system event log configuration.

Use the `show` form of this command to display the BMC system event log configuration.

Chapter 23. List of Acronyms

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

Acronym	Description
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
I/O	Input/Output
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IGMP	Internet Group Management Protocol
IGP	Interior Gateway Protocol
IPS	Intrusion Protection System
IKE	Internet Key Exchange
IP	Internet Protocol
IPOA	IP over ATM
IPsec	IP Security
IPv4	IP Version 4
IPv6	IP Version 6
ISAKMP	Internet Security Association and Key Management Protocol
ISM	Internet Standard Multicast
ISP	Internet Service Provider
KVM	Kernel-Based Virtual Machine
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
LDAP	Lightweight Directory Access Protocol
LLDP	Link Layer Discovery Protocol
MAC	medium access control
mGRE	multipoint GRE
MIB	Management Information Base
MLD	Multicast Listener Discovery
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
NBMA	Non-Broadcast Multi-Access
ND	Neighbor Discovery
NHRP	Next Hop Resolution Protocol
NIC	network interface card

Acronym	Description
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OSPFv2	OSPF Version 2
OSPFv3	OSPF Version 3
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PAT	Port Address Translation
PCI	peripheral component interconnect
PIM	Protocol Independent Multicast
PIM-DM	PIM Dense Mode
PIM-SM	PIM Sparse Mode
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PTMU	Path Maximum Transfer Unit
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RHEL	Red Hat Enterprise Linux
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
RP	Rendezvous Point
RPF	Reverse Path Forwarding
RSA	Rivest, Shamir, and Adleman
Rx	receive
S3	Amazon Simple Storage Service
SLAAC	Stateless Address Auto-Configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SPT	Shortest Path Tree
SSH	Secure Shell
SSID	Service Set Identifier

Acronym	Description
SSM	Source-Specific Multicast
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TBF	Token Bucket Filter
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service
TSS	TCP Maximum Segment Size
Tx	transmit
UDP	User Datagram Protocol
VHD	virtual hard disk
vif	virtual interface
VLAN	virtual LAN
VPC	Amazon virtual private cloud
VPN	virtual private network
VRRP	Virtual Router Redundancy Protocol
WAN	wide area network
WAP	wireless access point
WPA	Wired Protected Access