VYATTA, INC. | Vyatta System

Quick Start Guide

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

COPYRIGHT

PROPRIETARY NOTICES

RELEASE DATE: March 2012

DOCUMENT REVISION. R6.4 v01

RELEASED WITH: R6.4.0

PART NO. A0-0090-10-0026

Contents

Chapter 1: Welcome

Thank you for choosing the Vyatta system.

Vyatta has changed the networking world by developing the first commercially-supported, open-source networking, security, and service solution. providing an alternative to over-priced, inflexible products from proprietary vendors. Vyatta solutions offer industry-standard routing and management protocols, support for most commonly-used network interfaces, and configuration via command-line interface (CLI) or graphical user interface (GUI).

Vyatta delivers the features, performance, and reliability of an enterprise-class secure router with the added benefits of flexible deployment options—x86 hardware, blade servers, virtualization—freedom to integrate applications, and the economic advantages of commodity hardware and components.

This Guide

This document is intended to:

- Provide an overview of the deployment options, and help you determine the best way to deploy the Vyatta system in your environment
- Provide an overview of the user interface options available on the system
- Walk you through basic configuration of the system based on a sample scenario

Additional Resources

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

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

Chapter 2: Deploying the Vyatta System

The Vyatta system supports a number of flexible deployment options. This chapter describes the different platforms on which you can deploy the Vyatta system.

In this chapter:

Run from LiveCD

A LiveCD runs the Vyatta software on a RAM disk on the host machine. The system uses the RAM disk for writeable sections of the file system and uses an external system, such as a TFTP server or USB memory stick, for saving configuration files.

The LiveCD can run on a machine with an existing operating system without disturbing or changing the previously installed operating system. Configuration is not stored on the system, but you can save configuration to a persistent device such as a USB memory stick. Also, you cannot upgrade an installation run from LiveCD; each upgrade requires a fresh LiveCD. While you are running the system from a LiveCD, you will not be able to access other applications or programs on your machine.

A LiveCD is also required to install the Vyatta system to a persistent device, such as a hard disk.

As a method of deployment, running from LiveCD is best suited for evaluation and test scenarios.

You can read about using LiveCDs in *Installing the System: Using a LiveCD*.

Install on Physical Hardware

The Vyatta system can be installed and run on most standard x86 servers and PCs. The system can be installed from a LiveCD onto a variety of persistent devices, including hard drive, USB memory stick, and compact Flash.

You can install the system from a LiveCD you create beforehand. The install process uses the LiveCD as the source image, formats the device where the system is being installed, installs the system and, if possible, preserves configuration from previous installations. When the install process is complete, you reboot your system and the newly-installed system begins running.

NOTE *Not all hardware supports the ability to boot from USB device or Flash; check the BIOS of your hardware to see if yours does.*

You can read about installing onto a hard disk or other persistent device in *Installing and Upgrading: Hard Disks and Persistent Devices*.

Install into a Virtualized Environment or Cloud

The Vyatta system can be run as a software appliance in a virtual or cloud computing environment, allowing you to virtualize your network. The Vyatta software appliance has been optimized for a number of virtual environments. These platforms provide a great deal of flexibility as to how the virtual machine can be configured,

especially with respect to memory and Ethernet interfaces. Multiple Vyatta systems can be run simultaneously on a single hardware platform configured for multiple virtual machines.

The Vyatta software can be run in the following virtual environments:

- VMware
- XenServer
- RedHat KVM
- Amazon Machine Image

VMware

Like other virtualization platforms, VMware products provide the ability to run multiple virtual systems on a single hardware platform. The Vyatta system can be run on VMware ESX and ESXi. VMware ESX and ESXi are virtualization platforms that run directly on system hardware in a 64-bit environment.

For subscription customers, Vyatta provides a prebuilt VMware template that can be used to create Vyatta virtual machines on VMware ESX or ESXi. For community customers, Vyatta provides a virtualization ISO that can be installed on VMware ESX or ESXi.

You can read about installing onto VMware in *Installing and Upgrading: VMware*.

XenServer

Citrix XenServer is a server virtualization platform. Like other virtualization platforms, XenServer provides the ability to run multiple virtual machines on a single hardware platform.

XenCenter is the management system for XenServer. XenCenter allows you to manage multiple physical servers running XenServer, in addition to all the virtual machines running on each of the physical servers.

For subscription customers, Vyatta provides a prebuilt virtual machine template for XenServer that can be used by XenCenter to instantiate Vyatta virtual machine instances on XenServers. For community customers, Vyatta provides a virtualization ISO that can be used to create a Vyatta virtual machine on XenServer.

You can read about installing onto XenServer in *Installing and Upgrading: XenServer*.

RedHat KVM

The Vyatta system supports the Red Hat Kernel-Based Virtual Machine hypervisor on RHEL. Like other virtualization platforms, the Red Hat KVM provides the ability to run multiple virtual systems on a single hardware platform. Vyatta provides a prebuilt system image that runs on the KVM on RHEL.

This feature is available only in the Vyatta Subscription Edition.

You can read about installing onto RedHat KVM in *Installing and Upgrading: RedHat KVM*.

Amazon Machine Image

Amazon Web Services (AWS) is Amazon's cloud computing service. AWS provides the tools and infrastructure required by businesses to run compute environments "within the cloud."

At the core of AWS is the Amazon Machine Image (AMI). An AMI is a virtual machine template. You instantiate virtual machines instances from the template within the AWS cloud. A variety of AMIs are available from a number of vendors. The Vyatta AMI is a version of the Vyatta Subscription Edition system packaged to run in the AWS cloud.

This feature is available only in the Vyatta Subscription Edition.

You can read about installing and using a Vyatta AMI in AWS in *Installing the System/User Guide: Amazon Machine Interface*.

Chapter 3: Creating and Using a LiveCD

The quickest and easiest way to try out the Vyatta system is using a LiveCD. This chapter describes how to create and use a LiveCD.

In this chapter:

About LiveCDs

A LiveCD runs the Vyatta software on a RAM disk on the host machine. The system uses the RAM disk for writeable sections of the file system and uses an external system, such as a TFTP server or USB memory stick, for saving configuration files.

The LiveCD can run on a machine with an existing operating system without disturbing or changing the previously installed operating system. Configuration is not stored on the system, but you can save configuration to a persistent device such as a USB memory stick. Also, you cannot upgrade an installation run from LiveCD; each upgrade requires a fresh LiveCD. While you are running the system from a LiveCD, you will not be able to access other applications or programs on your machine.

A LiveCD is also required to install the Vyatta system to a persistent device, such as a hard disk.

As a method of deployment, running from LiveCD is best suited for evaluation and test scenarios.

Creating a LiveCD

The LiveCD must be bootable. See the documentation for your CD-burning utility for information on how to burn a bootable ISO image.

To create a bootable LiveCD

1. Download one of the ISO software images directly from the Vyatta web site.
   - Vyatta Subscription Edition customers use http://packages.vyatta.com/vyatta-supported/iso/stable/.
   - Vyatta community users use http://www.vyatta.org/downloads/.
2. Use CD-burning software to create a bootable ISO image. Note that:
   - The CD must be an ISO image: it won't work to just copy files onto the CD.
   - The CD must be bootable.

Specifying the Boot Device in the System BIOS

Insert the LiveCD into the CD drive of the system you wish to run it on. If your system is not already configured to boot from CD or DVD if one is present, you must do so in order to boot from the LiveCD.

To specify the boot device

1. During the boot sequence, press the appropriate key (for example, <F2>) to interrupt the boot sequence and enter your system's BIOS setup program.
2. In the boot sequence menu arrange the boot devices such that the device name for CD or DVD is first in the list. This will allow the system to boot from a CD or DVD if one is present.
3. Save the settings and reboot the system. When the system restarts, it will boot from the LiveCD.

Once the system has booted, you can confirm that you can access it from your network.

Testing the System

Once the system has successfully booted you will see the **vyatta login:** prompt. This indicates that the system is operational.

You should:

- Verify the Release and System Type
- Verify Connectivity

Verify the Release and System Type

The next step is to confirm that the correct release is running and it is running on the device that you expect.

To verify the release and system type

1. Login as user **vyatta** with password **vyatta** (default login ID).
2. Run the **show version** command, as in the following example. Example 3-1 shows version information for a system running on an Intel 32-bit hardware-based system.

Example 3-1 Displaying version information

```
vyatta@vyatta:~$ show version
Version:      VSE6.3-2011.07.21
Description:  Vyatta Subscription Edition 6.3 2011.07.21
Copyright:    2006-2011 Vyatta, Inc.
Built by:     autobuild@vyatta.com
Built on:     Thu Jul 21 06:05:29 UTC 2011
Build ID:     1107210624-d7a3790
```

```
System type:  Intel 32bit
Boot via:     image
HW model:     Latitude E6520
HW S/N:       9KC95P2
HW UUID:      43454D4C-4B00-1022-3454-B9B044382349
Uptime:      19:44:57 up 15 min,  1 user,  load average: 0.00, 0.12, 0.11
vyatta@vyatta:~$
```

The **Version:** line shows the version number of the running system. In this example the version is **VSE6.3-2011.07.21**.

The **System Type:** line shows the type of hardware the system is running on and whether it is in a virtual environment. The system in the example is running on an Intel 32-bit system, not in a virtual environment.

The **Boot via:** line shows the type of system that is running:

- **livecd** — The system is running from a LiveCD.
- **image** — The system is running as an image-based system.
- **disk** — The system is running as a disk-based system.

The system in the example is running as an image-based system.

Verify Connectivity

The final step is to confirm that the Vyatta system can be accessed on the local network. A quick and easy way to do this is to configure an Ethernet interface on the system and then ping the interface from another host on the network.

NOTE *Make sure that the system is physically connected to the network first.*

To test system connectivity

1 At the command prompt, enter the commands shown in the example, substituting an IP address on your existing subnet. In this example:

- The network is 192.168.1.0/24
- The IP address of the interface is 192.168.1.81

Make the appropriate substitutions for your network, as in the following example.

Example 3-2 Configuring a test Ethernet interface

```
vyatta@vyatta:~$ configure
vyatta@vyatta# set interfaces ethernet eth0 address 192.168.1.81/24
vyatta@vyatta# commit
vyatta@vyatta# exit
```

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

2 From another host on the same subnet, ping the interface to ensure that it is up. From a Linux or Windows command prompt, enter the following command (substituting the IP address you assigned to the interface):

```
ping 192.168.1.81
```

If the Vyatta system is reachable, you will see replies from it in response to the pings. If so, your system is installed and accessible on your network.

Chapter 4: The CLI and the Web GUI

There Vyatta system supports a rich and flexible command-line interface. The Vyatta Subscription Edition also supports a basic web GUI. This chapter provides a brief introduction to these two interfaces.

(The Vyatta Subscription Edition, it is also possible to remotely execute commands using the Vyatta Remote Access API. For more information about this, see the *Vyatta Remote Access API Reference Guide*.)

This chapter presents the following topics:

The Command-Line Interface (CLI)

This section presents the following topics:

The Vyatta CLI and the System Shell

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

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

All Vyatta users have access to operating system commands as well as Vyatta CLI commands. The amount of access to the operating system commands varies with the privilege level of the user.

The operating system commands and constructs are available from any point within the Vyatta CLI.

Command Modes

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

- Operational mode provides access to operational commands for showing and clearing information and enabling or disabling debugging, as well as commands for configuring terminal settings, loading and saving configuration, and restarting the system.
- Configuration provides access to commands for creating, modifying, deleting, committing and showing configuration information, as well as commands for navigating through the configuration hierarchy.

When you log on to the system, the system is in operational mode.

- To enter configuration mode from operational mode, issue the **configure** command.

- To return to operational mode from configuration mode, issue the **exit** command. If there are uncommitted configuration changes, you must either commit the changes using the **commit** command, or enter **exit discard** to discard the changes before you can exit to operational mode.

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

▶ Try it Enter configuration mode

In configuration mode you can **set**, **delete**, and **show** information. Enter configuration mode by typing **configure** at the command prompt in operational mode.

```
vyatta@vyatta:~$ configure
[edit]
vyatta@vyatta#
```

Notice how the command prompt changes to remind you what mode you are in.

▶ Try it Exit configuration mode

Exiting configuration mode returns you to operational mode.

```
vyatta@vyatta# exit
exit
vyatta@vyatta:~$
```

CLI Help

You can get CLI help by entering any of the following at the command prompt:

Type this:	To see this:
help	Displays available system shell commands
help *command*	Help and usage guidelines for the specified shell command.
<Tab> ?	For non-root users, displays all available Vyatta commands and provides command completion. For the root user,<Tab> displays all available Vyatta and shell commands and provides command completion, however, “?” will not work.

▶ Try it **Find available commands**

At the command prompt in operational mode, press the <Tab> key or the "?" key.

```
vyatta@vyatta:~$ <Tab>
add         copy        generate    reboot      restart     traceroute
clear       delete      install     release     set         update
clone       disconnect  monitor     rename      show        upgrade
configure   force       ping        renew       telnet
connect     format      poweroff    reset       test
vyatta@vyatta:~$
```

Command Completion

Unlike some proprietary router CLIs that accept an unambiguous prefix, the Vyatta system requires that you type the full command name. Thus, command completion is very important for usability. As an example, Cisco allows you to type **sh config**. The Vyatta system would require you to type **show configuration**, but you can get the same effect in the Vyatta system with **sh**<Tab>**config**<Tab> as the <Tab> completes the unambiguous command.

Pressing the <Tab> key provides command completion. You can use the <Tab> to show:

- All available commands
- All valid completions for a command partially typed in
- The next available set of options for a command. You can use the <Tab> key iteratively in this way to discover complete command syntax.

NOTE *The "?" key, like the <Tab> key, will provide help strings for commands.*

▶ Try it **Use command completion on an unambiguous command**

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

```
vyatta@vyatta:~$ sh<Tab>
vyatta@vyatta:~$ show
```

▶ Try it **Use command completion on an ambiguous command**

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

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

▶ Try it Display help strings for commands

The following example requests command completion for the typed string **s** a second time. In this case, help strings for the possible completions are displayed.

```
vyatta@vyatta:~$ s<Tab>
Possible completions:
   set       Set system or shell options
   show      Show system information

vyatta@vyatta:~$ s
```

Viewing Long Output (“More”)

If the information being displayed is too long for your screen, the screen shows a line number indication where the information breaks. The Linux operating system provides lots of commands for controlling information in a “More” display; a few important ones are shown below. (See Linux documentation for additional control of “More” displays.)

To do this	Press this
Exit “More”	q
Scroll down one whole screen.	<Space>
Scroll up one whole screen.	b
Scroll down one line.	<Enter>
Scroll up one line.	<Up Arrow>

▶ Try it Show system configuration

Entering **show** in configuration mode shows information you’ve explicitly set. Entering show -all shows information you’ve set plus all default information.

Enter configuration mode and enter **show -all** at the command prompt. The full default system displays and the "More" prompt displays. (Some lines are left out of the example to save space.)

```
vyatta@vyatta:~$ configure
[edit]
vyatta@vyatta# show -all
...
    ethernet eth3 {
        duplex auto
        hw-id 00:14:bf:5a:84:f9
        speed auto
    }
...
:
```

▶ Try it Exit from a "More" display

Within the "More" display, enter **q**. You are returned to the command prompt.

```
    ethernet eth3 {
        address 192.168.1.85/24
        duplex auto
        hw-id 00:14:bf:5a:84:f9
        speed auto
    }
: q
[edit]
vyatta@vyatta#
```

Showing Support Information

If you need to report a bug or request support, you will need to supply version information for your software. You can do this in operational mode.

▶ Try it Show support information

If you are in configuration mode, return to operational mode.

```
vyatta@vyatta# exit
exit
```

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

Use the **show version** command to display version information.

```
vyatta@vyatta:~$ show version
Version  :    3.0.2
Copyright:    2006-2008 Vyatta, Inc.
Built by :    autobuild@vyatta.com
Built on :    Wed Apr 16 08:26:33 UTC 2008
Build ID :    080416082620a705a
Boot via :    livecd
Uptime   :   14:22:45 up 35 min,  2 users,  load average: 0.00, 0.00, 0.00
vyatta@vyatta:~$
```

The Web GUI

The web GUI is available only in the Vyatta Subscription Edition.

The web GUI is an alternative user interface for interacting with the Vyatta system. For security reasons, the web GUI is turned off by default. If you want to use the web GUI, you must enable it through the CLI.

Any operation that can be performed through the CLI (except enabling the GUI) can also be performed through the web GUI. The web GUI essentially reflects the structure of the CLI; in particular, the command hierarchy in the GUI follows the basic CLI configuration structure. If you are familiar with the CLI, the structure of the GUI should be straightforward to understand.

Supported browsers include Firefox 3, Internet Explorer 7 and 8, and Google Chrome 5.

This section presents the following topics:

- Configuration
- Operation

Enabling Access to the Web GUI

All you need to do to access the web GUI is to enable the HTTPS service. (Note that enabling HTTPS access only allows access to the web GUI; it does not provide general SSL access.)

To access the web GUI from the network, you will also need to configure an Ethernet interface with an IP address. In our examples, we assume you have configured an Ethernet interface with an IP address.

▶ Try it Enable web GUI access

1 In configuration mode, enable HTTPS access to the GUI on the Vyatta system and commit the change.

```
vyatta@R1# set service https
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

Logging On to the Web GUI

For security purposes, all communication to the GUI is over HTTPS, the secure version of HTTP, using a self-signed certificate. As with any self-signed certificate, when you initially connect with a web browser you may receive a warning that the certificate is not trusted. Since this certificate is generated on the Vyatta system specifically for browser access, this message can be safely ignored and the certificate stored for future access.

▶ Try it Access and Log On to the GUI

1 Point your web browser at the IP address defined for the Ethernet interface. (Prefixing the IP address with **https://** is optional.) The login screen opens in your browser.

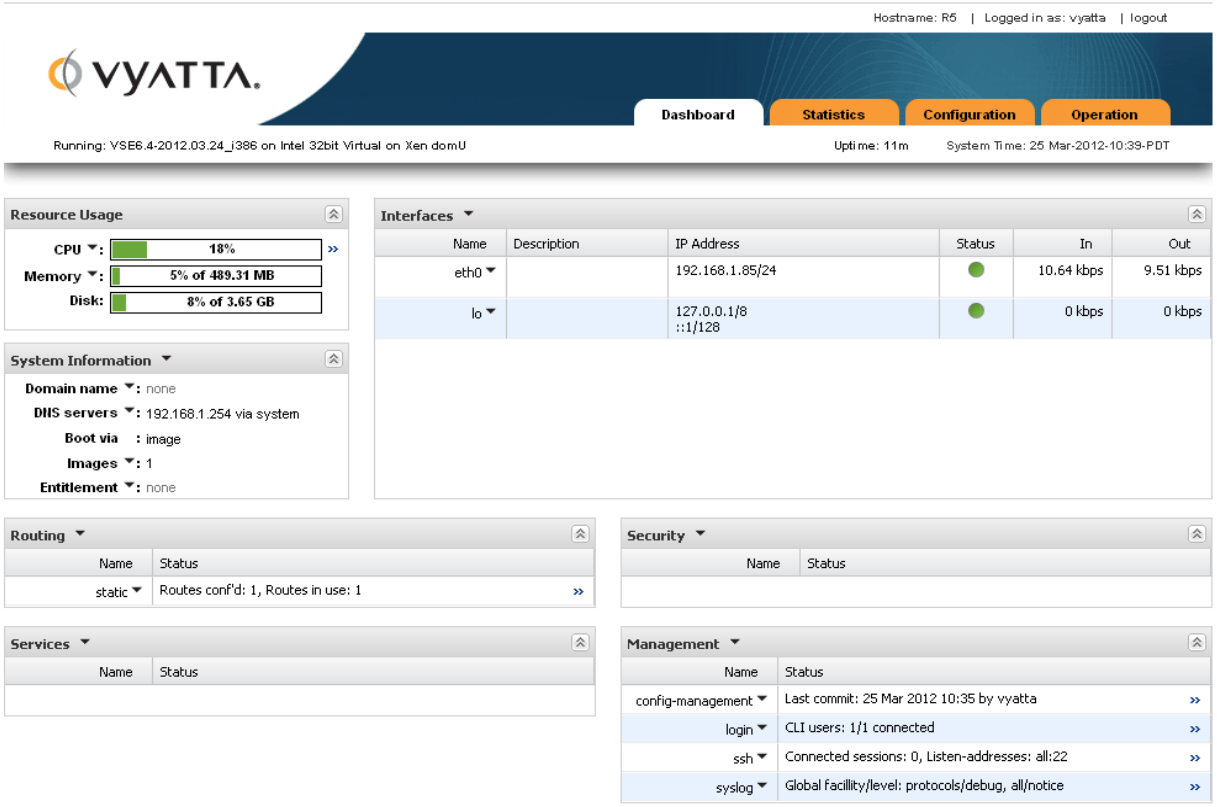
2 Log on. Any non-root user may log on to the router through the GUI. Root login through the GUI is not supported for security reasons. The default non-root user ID and password are as follows:

```
User name: vyatta
Password: vyatta
```

Passwords are independent of the means of access: the CLI and the GUI use the same login credentials.

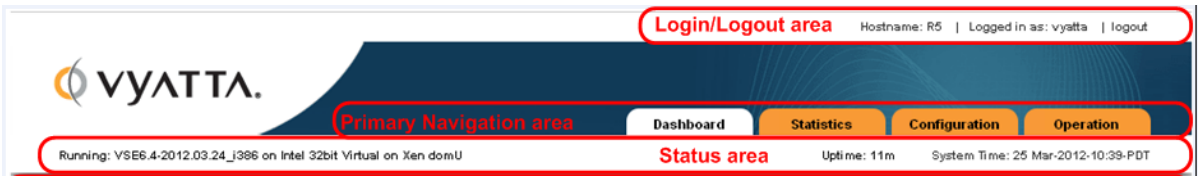
NOTE *For security reasons, it is strongly recommended that each user accessing the system from a web browser have an individual user account.*

Once you have logged on, the web GUI opens to its dashboard.



Basic Navigation in the Web GUI

On the Dashboard screen, take note of three areas of the screen that are common to all web GUI pages: the Login/Logout area, the primary navigation area, and the status area. These areas occur on all web GUI screens.



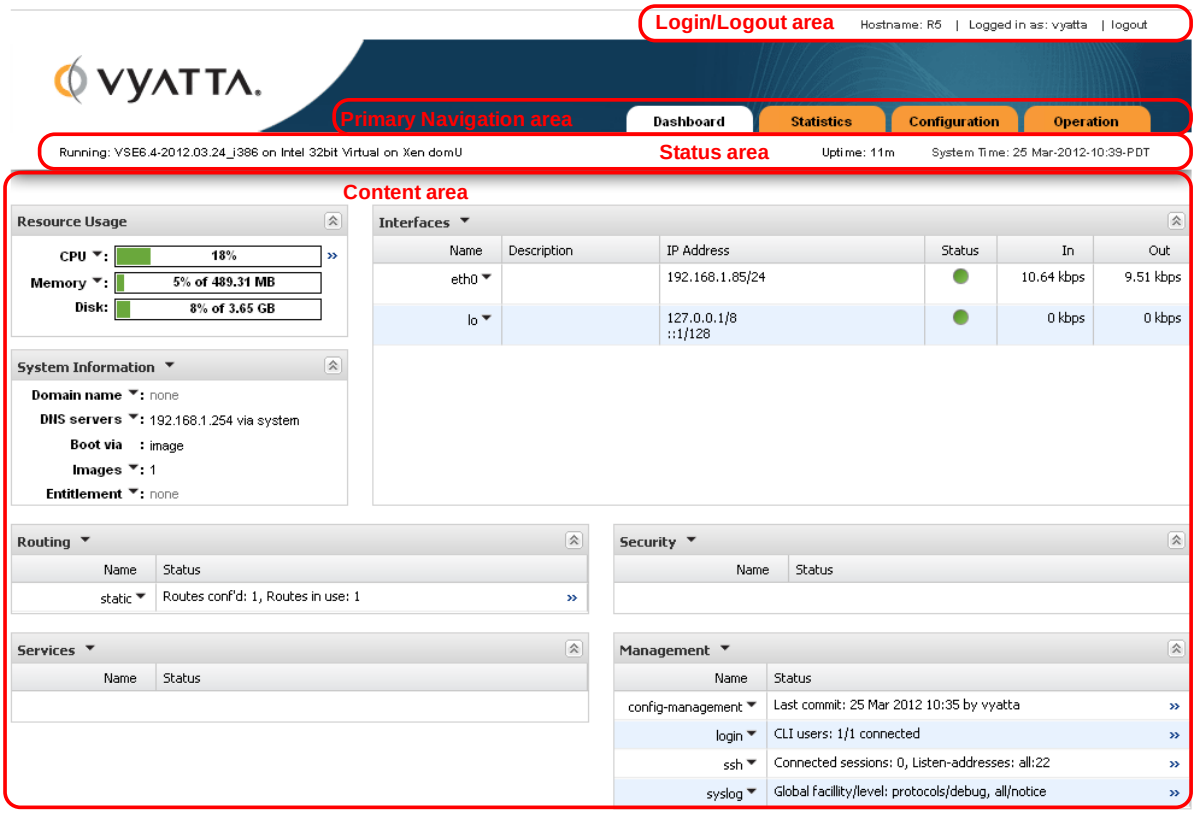
The **Login / Logout area** provides information about the system being accessed and the username of the user who is logged on. There is also a "logout" link you can use to log off the system.

The **primary navigation area** provides tabs to allow you to move back and forth between the main areas of the GUI: **Dashboard**, **Statistics**, **Configuration**, and **Operation**.

The **status area** shows you the version of the system you are running, the amount of time the system has been up, and the system time.

Dashboard

The **content area** of the Dashboard displays operational status and configuration for key areas of the system.



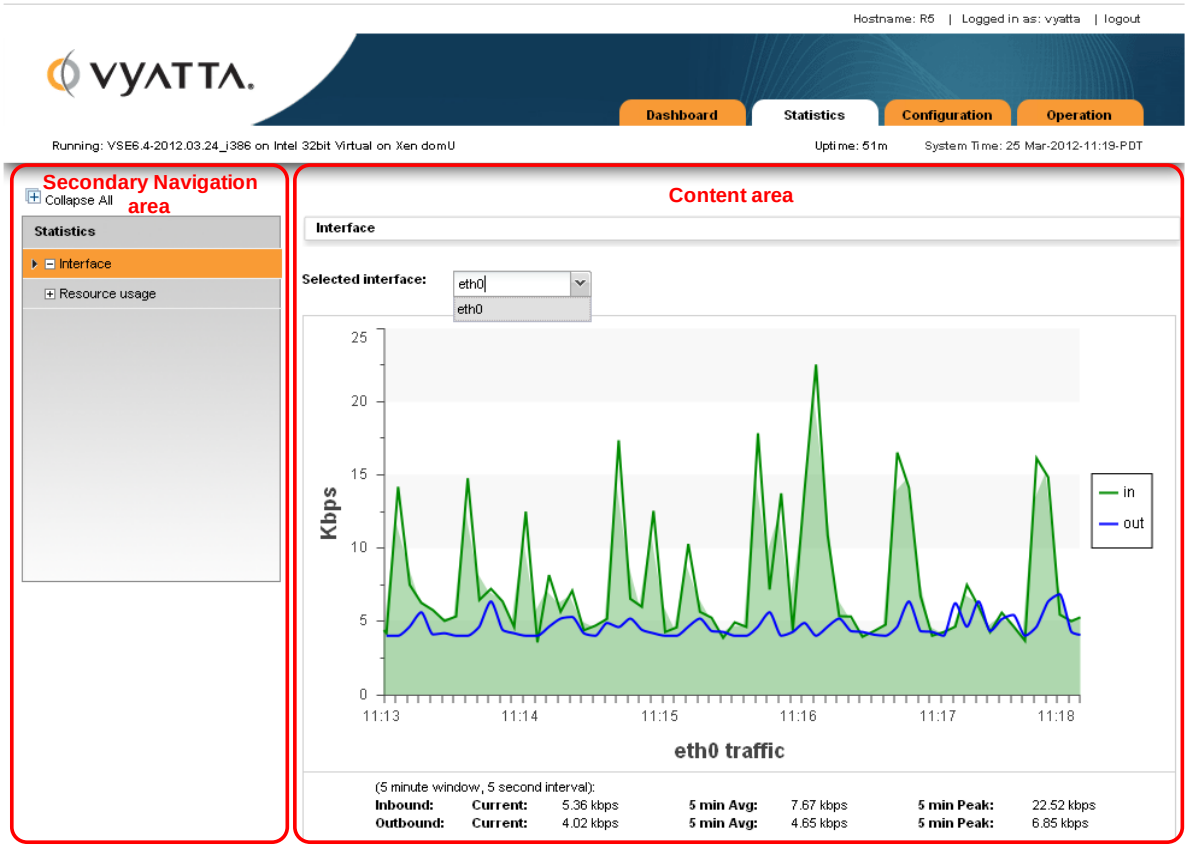
Statistics

The Statistics screen shows a visual representation of ongoing system statistics you select.

The **secondary navigation area** allows you to navigate to the system component you want to see statistics for.

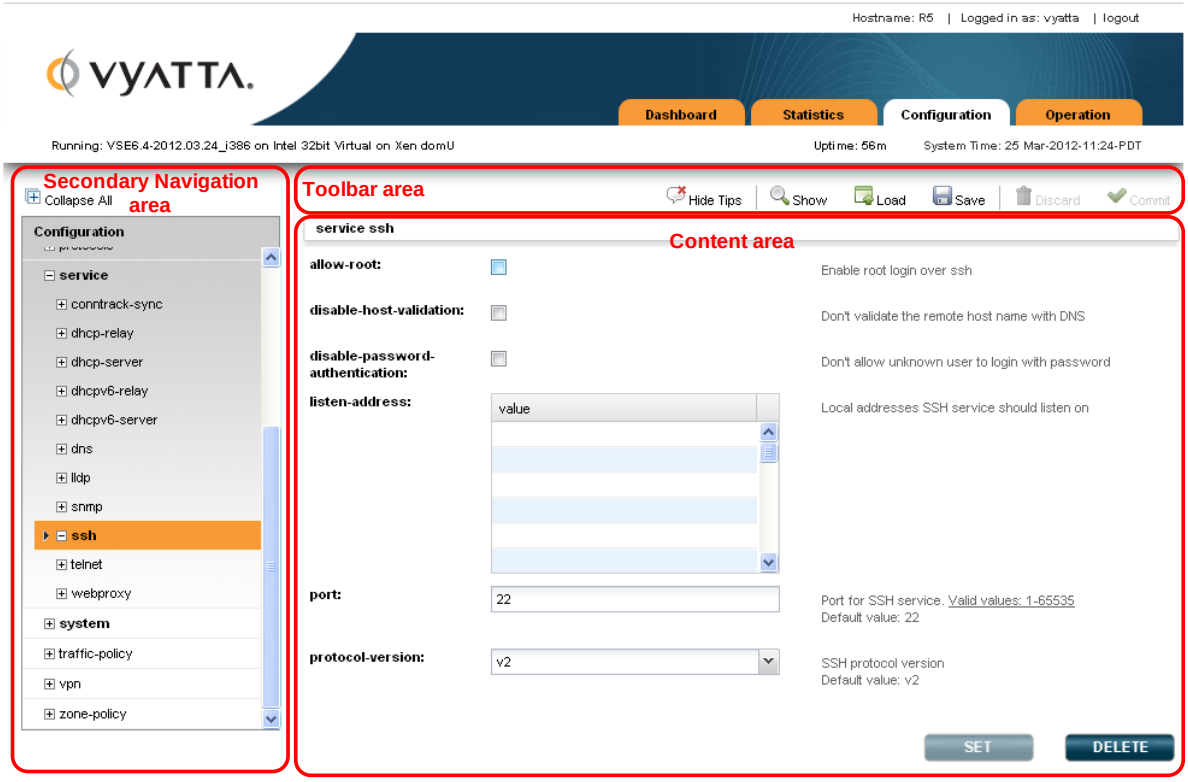
The **content area** shows real-time statistics for the selected component.

In the Statistics screen below, statistics are being displayed for the eth0 interface.



Configuration

The Configuration screen allows you to configure system components. The configuration options available on this screen are the same as those available in the Vyatta CLI. To learn about these options, refer to Vyatta documentation for the CLI.



The **toolbar area** of the Configuration screen provides access to tools available for a given configuration command. The following buttons are available on the toolbar:

- **Hide Tips / Show Tips**: Toggles between showing and hiding help tips within the Content area.
- **Show**: Displays the proposed configuration. New or modified fields are indicated with a "+". Deleted fields are indicated with a "-". Once the proposed configuration is committed and becomes the active configuration, these indicators are removed.
- **Load**: Loads the specified configuration to become the active configuration. The configuration file specified can be local or remote.
- **Save**: Saves the active configuration. The file can be stored either locally or remotely.
- **Discard**: Discards any changes (inidcated by yellow dots) that have been made prior to **Commit** being pressed.
- **Commit**: Commits changes to the active configuration.

The **secondary navigation area** allows you to navigate to the component you want to configure. As you progress down the navigation hierarchy, control is passed to the appropriate level within the hierarchy.

The **content area** is where configuration is modified. The following buttons are available in context within the content area:

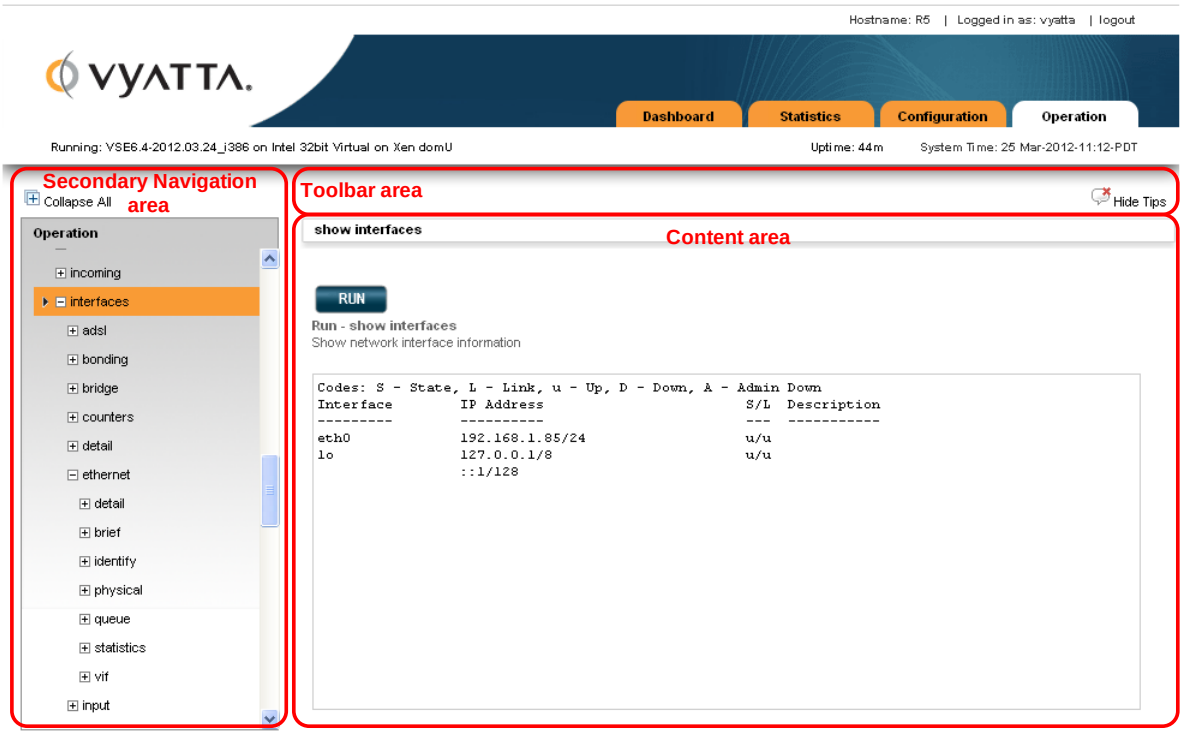
- **Set**: Confirms changes made to the current screen. Pressing **Commit** activates the changes.
- **Delete**: Removes the selected configuration node. Pressing **Commit** activates the change.
- **Create**: Creates a new configuration node. Pressing **Commit** activates the change.
- **Add**: Adds an entry to a multi-value leaf node. Pressing **Commit** activates the change.
- **Trash can**: Removes an entry from a multi-value leaf node. Pressing **Commit** activates the change.

Various indicators are used to provide information regarding commands and configuration fields. These are as follows:

- **Bold text in the hierarchy**: Indicates that the node is currently configured on the system.
- **Plain text in the hierarchy**: Indicates that the node is not currently configured in the system but is available for configuration.
- **Red asterix**: Indicates that a field is a required field.
- **Yellow dot**: Indicates that the configuration has been modified. A "-" inside the dot indicates that the node is to be deleted. Pressing **Commit** activates the change.
- **Red dot**: Indicates that the configuration is in error and must be changed before it will be accepted.

Operation

The Operation screen allows you to run operational commands. These are the same commands that are available in operational mode within the CLI. To learn about these commands, refer to Vyatta documentation for the CLI.



The **secondary navigation area** at left allows you to navigate among the available operational commands.

The **content area** is where commands are run and output, if any, is displayed.

The following buttons are available within the content area of the Operation screen:

- **Run**: Runs the selected command.
- **Stop**: Stops the output for commands that generate output.
- **Pause**: Pauses the output for commands that generate output.

The Operation screen also has a **toolbar**. The following button is available within the toolbar:

- **Hide Tips / Show Tips**: Toggles between showing and hiding help tips within the content area.

Chapter 5: Quick Start Configuration Scenarios

This chapter introduces you to basic configuration concepts on the Vyatta CLI, and then steps you through two configuration scenarios:

- Some basic system configuration tasks
- A scenario with the Vyatta system acting as an Internet gateway

The examples in this chapter use the Vyatta CLI and assume that the Vyatta system is installed on a hard disk.

This chapter presents the following topics:

Configuration Basics in the CLI

This section presents the following topics

Configuration Hierarchy

From the system's point of view, a configuration *leaf node* is different from a simple configuration *non-leaf node*. A configuration leaf node takes the form *node value*, as in the following example.

```
protocol-version v2
```

A non-leaf node always has an enclosing pair of braces, which may be empty, as in the following example.

```
loopback lo {
}
```

or non-empty, as in the following example.

```
ssh {
    port 22
    protocol-version v2
}
```

Adding and Modifying Configuration

Add new configuration by creating a configuration node, using the **set** command in configuration mode. Modify existing configuration also by using the **set** command in configuration mode.

▶ Try it Add configuration

In configuration mode, set the IP address of Ethernet eth0 interface using the **set** command.

```
vyatta@vyatta# set interfaces ethernet eth0 address 192.0.2.21/24
[edit]
vyatta@vyatta#
```

Note that the configuration node for interface eth0 will already exist, assuming that your system has at least one Ethernet. That's because the system automatically discovers physical interfaces on startup and creates configuration nodes for them. For the same reason, the hardware ID (MAC address) of interface eth0 will also be known to the system.

Now use the **show** command to see the addition.

```
vyatta@vyatta# show interfaces ethernet eth0
+address 192.0.2.21/24
 hw-id 00:13:46:e6:f6:87
[edit]
vyatta@vyatta#
```

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

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.

▶ Try it **Modify configuration**

For the most part, modifying configuration is the same as adding configuration by using the **set** command. This works for identifiers of nodes containing a single instance. To change the identifier of a node for which there can be multiple instances (a "multi-node"), such as a DNS server or an IP address for an interface, you must delete the node and recreate it with the correct identifier.

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

Deleting Configuration

You delete configuration statements, or complete configuration nodes, using the **delete** command.

▶ Try it **Delete configuration**

Delete a configuration node:

```
vyatta@vyatta# delete interfaces ethernet eth0 address 192.0.2.21/24
[edit]
vyatta@vyatta#
```

Now use the **show** command to see the deletion.

```
vyatta@vyatta# show interfaces ethernet eth0
-address 192.0.2.21/24
 hw-id 00:13:46:e6:f6:87
[edit]
vyatta@vyatta#
```

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

Some configuration nodes 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.

Committing Configuration Changes

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

Uncommitted changes are flagged with a plus sign (for additions), a greater-than sign (for modifications), or a minus sign (for deletions).

```
vyatta@vyatta# show interfaces ethernet eth0
-address 192.08.2.21/24
 hw-id 00:13:46:e6:f6:87
[edit]
vyatta@vyatta#
```

▶ Try it **Commit configuration changes**

Commit any uncommitted changes by issuing the **commit** command in configuration mode.

Once you commit the changes, the indicator disappears. Also note that the non-mandatory configuration node (**address**) is removed from the configuration.

```
vyatta@vyatta# commit
[edit]
vyatta@vyatta# show interfaces ethernet eth0
 hw-id 00:13:46:e6:f6:87
[edit]
vyatta@vyatta#
```

Discarding Configuration Changes

Instead of deleting many specific changes, you can abandon all changes made within a configuration session by using the **discard** command or by using the **exit** command with the **discard** option.

▶ Try it **Discard configuration changes using "discard"**

Using the **discard** command is the same as deleting all changes made within a configuration session and then committing the changes. Also notice that after the **discard** command has completed you stay in configuration mode.

```
vyatta@vyatta# show interfaces ethernet eth0
+address 192.0.2.21/24
+description “This is a test”
 hw-id 00:13:46:e6:f6:87
[edit]
vyatta@vyatta# discard
Changes have been discarded
[edit]
vyatta@vyatta# show interfaces ethernet eth0
 hw-id 00:13:46:e6:f6:87
[edit]
vyatta@vyatta#
```

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

▶ Try it Discard configuration changes using “exit discard”

Try exiting from configuration mode with uncommitted configuration changes; you won’t be able to. Discard the changes by issuing the **exit discard** command.

```
vyatta@vyatta# exit
Cannot exit: configuration modified.
Use 'exit discard' to discard the changes and exit.
[edit]
vyatta@vyatta# exit discard
exit
vyatta@vyatta:~$
```

Saving Configuration

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

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

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.

NOTE *Unless you save your configuration changes to the default configuration file, they do not persist when the system is restarted. On restart, the configuration is loaded from the* ***config.boot*** *file.*

▶ Try it Save configuration to the default configuration file

To save to the **config.boot** file in the default configuration directory, just enter **save** in configuration mode.

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

▶ Try it Saving configuration to another file

Save to a different file name in the default directory by specifying a different file name.

```
vyatta@vyatta# save testconfig
Saving configuration to '/config/testconfig'...
Done
[edit]
vyatta@vyatta#
```

This provides the ability to have multiple configuration files for different situations (for example, test and production).

You can also save a configuration file to a location path other than the standard configuration directory **/config**, by specifying a different path. You can save to a hard drive, compact Flash, or USB device by including the directory the device is mounted on in the path.

If you are running the system from LiveCD, configuration can be saved only to floppy disk (**/media/floppy/config/config.boot**). If you do not save your running configuration to floppy disk, any changes are lost on reboot.

Before saving to floppy disk, you must initialize the floppy disk for use on the system.

▶ Try it **Initialize a floppy disk for configuration files**

The floppy drive is referred to as **/dev/fd0**. It is automatically mounted in the directory **/media/floppy**.

1. Insert a blank floppy disk into the floppy disk drive.
2. At the system command prompt, enter the following:

```
vyatta@vyatta:~$ init-floppy
```

The system prepares the floppy to receive configuration files. It also saves a copy of the current configuration to **/media/floppy/config/config.boot**.

▶ Try it **Save LiveCD configuration to the default location on floppy disk**

If you want to be able to boot from the configuration file, save configuration to **/media/floppy/config/config.boot**.

```
vyatta@vyatta# save
Saving configuration to '/media/floppy/config/config.boot'...
Done
[edit]
vyatta@vyatta#
```

▶ Try it **Save LiveCD configuration to another file on floppy disk**

If you want to save a record of configuration, save to a different file name in **/media/floppy/config**.

```
vyatta@vyatta# save testconfig1
Saving configuration to '/media/floppy/config/testconfig1'...
Done
[edit]
vyatta@vyatta#
```

Loading Configuration

A configuration can be loaded using the **load** command in configuration mode. You can only load a file that has first been saved using the **save** command.

▶ Try it **Load configuration from the default directory**

Load a configuration file from the default directory by specifying only the file name.

```
vyatta@vyatta# load testconfig
Loading configuration file /config/testconfig...
No configuration changes to commit
Done
[edit]
vyatta@vyatta#
```

To load from a directory other than the default directory the full path must be specified.

Changing the Default Configuration File

To change the default configuration file to one that you have previously saved you use both the **load** and **save** commands.

▶ Try it **Load a previously saved configuration and save it as the default**

Load the previously saved configuration file (**testconfig**) from the default directory by specifying only the file name and then save it to the default (**config.boot**).

```
vyatta@vyatta# load testconfig
Loading configuration file /config/testconfig...
No configuration changes to commit
Done
[edit]
vyatta@vyatta# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyatta@vyatta#
```

At this point the configurations in **testconfig** and **config.boot** are the same. The currently active configuration is the same as what is loaded when the system restarts.

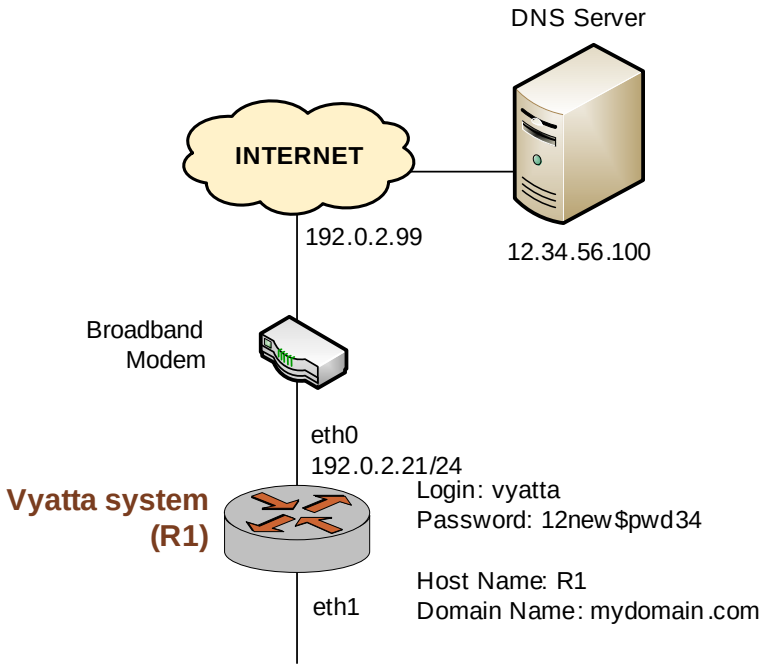
Scenario: Basic System Configuration

This section steps you through initial system configuration tasks using the CLI . These are tasks that are required for almost any scenario in which you might use the Vyatta system. These include the following:

Overview

Figure 5-1 shows a network diagram of the basic system configuration scenario.

Figure 5-1 Scenario: Basic System Configuration



This section presents the following topics:

Logging On

The first step is to log on. Our examples use the predefined user **vyatta**.

▶ Try it **Log on**

Log on as user **vyatta**. The default password for this user is **vyatta**. The password is not echoed onto the screen.

```
Welcome to Vyatta - vyatta tty1
vyatta login: vyatta
Password:
Linux vyatta 2.6.20 #1 SMP Fri Sep 21 02:22:08 PDT 2007 i686
Welcome to Vyatta.
This system is open-source software. The exact distribution terms for each
module comprising the full system are described in the individual files
in /usr/share/doc/*/copyright.
Last login: Sat Nov 10 16:48:48 2007 on tty1
vyatta@vyatta:~$
```

Entering Configuration Mode

When you log on, you are in operational mode. To configure the system, you must enter configuration mode.

▶ Try it **Enter configuration mode**

Enter configuration mode by entering **configure**.

```
vyatta@vyatta:~$ configure
[edit]
vyatta@vyatta#
```

Notice how the command prompt changes to mark the move from operational mode (":~$") and configuration mode ("#").

Setting the Host Name

The default host name for a Vyatta device is **vyatta**. You can change this to fit in with your environment. In our example we use a host name of R1.

▶ Try it Set the host name

Change the host name to R1 using the **set system host-name** command. Remember to commit all configuration changes.

```
vyatta@vyatta# set system host-name R1
[edit]
vyatta@vyatta# commit
[edit]
vyatta@vyatta#
```

The command prompt changes to reflect the new host name the next time you log in.

Setting the Domain Name

In addition to changing the host name, you must specify the system's domain name. In our examples we use **mydomain.com** as the domain name.

▶ Try it Set the domain name

Set the domain name using the **set system domain-name** command.

```
vyatta@R1# set system domain-name mydomain.com
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

Changing Passwords

The Vyatta system has one pre-defined user account:

```
user ID: vyatta      password: vyatta
```

To secure your system, you should change the password for this user account.

▶ Try it **Change your password**

Change the password of user **vyatta** to **12new$pwd34** using the **set system login user** command, as follows:

```
vyatta@R1# set system login user vyatta authentication plaintext-password
12new$pwd34
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

Configuring Interfaces

The kind and number of interfaces you configure will depend on your physical device and the topology of your network. However, almost every topology will require that at least one Ethernet interface configured.

The kind and number of interfaces you can configure depends on your physical device. The Vyatta system automatically discovers all physical interfaces on startup and creates configuration nodes for them.

In this basic scenario, we'll configure the Ethernet interface eth0 as an Internet-facing interface. This will allow the system to reach the DNS server and default gateway configured in subsequent steps.

The loopback interface, which is a software interface, is also automatically created on startup, preconfigured to IP address 127.0.0.1/8. The loopback interface is always be available as long as the device is reachable at all. This makes the loopback interface particularly useful for mapping to the system host name, as a router ID in routing protocols such as BGP and OSPF, or as a peer ID for internal BGP peers.

▶ Try it **Configure an Internet-facing Ethernet interface**

Configure interface eth0 with an IP address of 192.0.2.21 and a prefix length of 24.

```
vyatta@R1# set interfaces ethernet eth0 address 192.0.2.21/24
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

NOTE *If your ISP provides a dynamic IP address, rather than the static we have used in the example, the command you would use would be* ***set interfaces ethernet eth0 address dhcp.***

To view the configuration, use the **show** command:

```
vyatta@R1# show interfaces
ethernet eth0 {
    address 192.0.2.21/24
    hw-id 00:40:63:e2:e4:00
}
loopback lo {
}
[edit]
vyatta@R1#
```

Configuring Access to a DNS server

In order to be able to translate host names (such as www.vyatta.com) to IP addresses (such as 76.74.103.45), the system must be able to access a DNS server.

▶ Try it Specify a DNS server

In our example, the DNS server is at IP address 12.34.56.100. Add the DNS server using the **set system name-server** command.

```
vyatta@R1# set system name-server 12.34.56.100
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

Specifying a Default Gateway

When the local system does not know what route to use for a given destination, it forwards packets to the “default gateway.” In our example, the ISP’s gateway at 192.0.2.99 acts as the default gateway.

▶ Try it

Specify the default gateway

Add the default gateway using the **set system gateway-address** command.

```
vyatta@R1# set system gateway-address 192.0.2.99
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

This completes basic system configuration using the CLI.

Scenario: Internet Gateway

This configuration scenario builds on the Basic System configuration scenario and steps through the process of configuring the system as a basic Internet gateway. The goal is for the system to provide the following:

1. The ability to route traffic between the office LAN and the Internet.
2. The ability for users to access the system from the local network using SSH.
3. DHCP capability for providing dynamic IP addresses to internal devices.
4. NAT capability for translating multiple internal addresses to a single external address.
5. Firewall capability for preventing system access from the Internet.

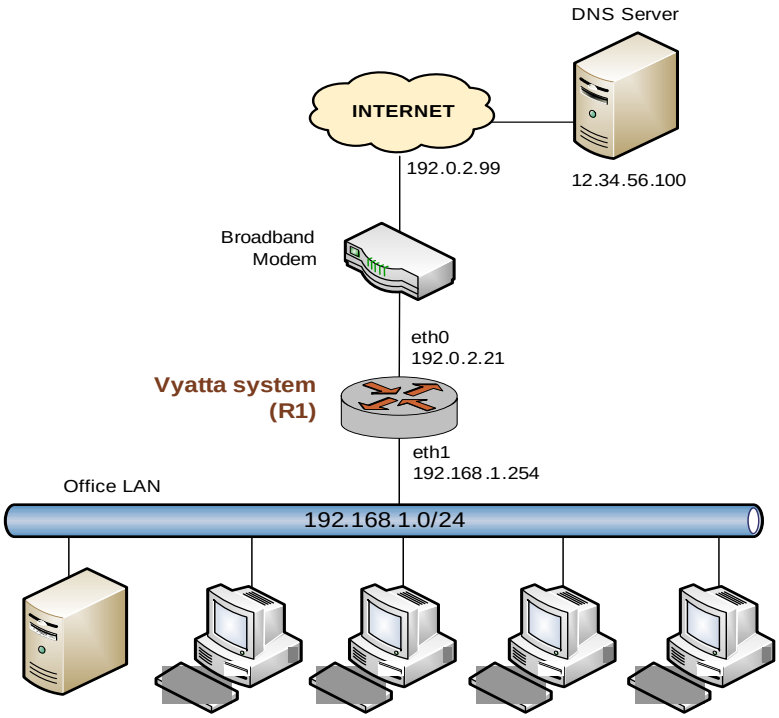
This section presents the following topics:

Overview

The example assumes a system with two Ethernet interfaces: one interface facing the office LAN and the other facing the Internet.

Figure 5-2 shows a network diagram of this sample scenario.

Figure 5-2 The Vyatta system configured as an Internet gateway



The examples in this scenario assume that you have completed the configuration shown in the basic scenario (see page 36).

Configuring Interfaces

In the basic scenario, we configured an Internet-facing Ethernet interface. To act as an Internet gateway, the system needs an additional Ethernet interface facing the office LAN. We'll use interface eth1 for this.

▶ Try it

Configure the office LAN–facing Ethernet interface

Assign IP address 192.168.1.254 to interface eth1 with a prefix length of 24, which is the prefix length of the office subnet.

```
vyatta@R1# set interfaces ethernet eth1 address 192.168.1.254/24
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

Let's look at the interfaces configured now:

```
vyatta@R1# show interfaces
ethernet eth0 {
    address 192.0.2.21/24
    hw-id 00:40:63:e2:e4:00
}
ethernet eth1 {
    address 192.168.1.254/24
    hw-id 00:13:46:e6:f6:87
}
loopback lo {
}
[edit]
vyatta@R1#
```

Enabling SSH Access

The gateway's SSH service should allow users to access the from the office LAN but not from the Internet. In this step, we enable the SSH service on the system to be accessed by using an SSH client. (Preventing access from the Internet will be done later, using the Vyatta system's firewall.)

(Note that setting up Telnet or web GUI access is done in a similar way as for SSH: by issuing the **set service telnet** or **set service https** commands, respectively.)

▶ Try it Enable SSH access

To allow SSH access, you enable the SSH service on the system. By default the system is set to use the more secure SSH version 2.

```
vyatta@R1# set service ssh
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

Configuring DHCP Server

DHCP provides dynamic IP addresses to hosts on a specified subnet. In our scenario, the DHCP server provides addresses to hosts on the office LAN (attached to interface eth1).

▶ Try it Set up DHCP Server

For the DHCP server, define an address pool from 192.168.1.100 to 192.168.1.199 to dynamically assign addresses to hosts on the office LAN. Also, set the default router and DNS server to the values that will be assigned to hosts on the office LAN. The default router for these devices will be the LAN-facing interface of the Internet gateway.

```
vyatta@R1# set service dhcp-server shared-network-name ETH1_POOL subnet
192.168.1.0/24 start 192.168.1.100 stop 192.168.1.199
[edit]
vyatta@R1# set service dhcp-server shared-network-name ETH1_POOL subnet
192.168.1.0/24 default-router 192.168.1.254
[edit]
vyatta@R1# set service dhcp-server shared-network-name ETH1_POOL subnet
192.168.1.0/24 dns-server 12.34.56.100
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

Let's view this configuration.

```
vyatta@R1# show service dhcp-server
    shared-network-name ETH1_POOL {
        subnet 192.168.1.0/24 {
            start 192.168.1.100 {
                stop 192.168.1.199
            }
            dns-server 12.34.56.100
            default-router 192.168.1.254
        }
    }

[edit]
vyatta@R1#
```

Configuring NAT

The Internet gateway should send outbound traffic from the office LAN out through the Internet-facing interface translating all internal private IP addresses to a single public address. This is done by defining a Network Address Translation (NAT) rule.

▶ Try it Define a NAT rule

Define a rule that allows traffic from network 192.168.1.0/24 to proceed to the Internet through interface eth0, and translates any internal addresses to eth0's IP address. (This is called "masquerade" translation.)

```
vyatta@R1# set service nat rule 1 source address 192.168.1.0/24
[edit]
vyatta@R1# set service nat rule 1 outbound-interface eth0
[edit]
vyatta@R1# set service nat rule 1 type masquerade
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

Let's view this configuration.

```
vyatta@R1# show service nat
    rule 1 {
        type masquerade
        outbound-interface eth0
        source {
            address 192.168.1.0/24
        }
    }

[edit]
vyatta@R1#
```

Configuring Firewall

As it is shipped, the Vyatta System does not restrict traffic flow. That is, unless a firewall rule is applied to an interface, the interface allows all traffic through it. The firewall functionality provides packet filtering, providing flexibility in restricting traffic as required.

In this simple scenario, the Internet gateway should allow hosts on the local network and services on the gateway itself to initiate traffic to the Internet, but it should drop all traffic initiated from the Internet. This section sets up a basic firewall configuration to do this.

Essentially, this sequence defines a firewall rule set allowing traffic initiated from, or passing through, the gateway to the Internet. All other packets are denied because there is an implicit **deny all** rule at the end of every firewall rule set.

In general, to configure a firewall on an interface:

1. You define a number of named firewall rule sets, each of which contains one or more firewall rules.

   When applying a firewall rule set, keep in mind that after the final user-defined rule, an implicit rule of **deny all** takes effect.

2. You apply the each of the named rule sets to an interface as a filter. You can apply one named rule set for each of the following on an interface:

   - **in.** If you apply the rule set to an interface as **in**, the rule set filters packets entering the interface.
   - **out.** If you apply the rule set to an interface as **out**, the rule set filters packets leaving the interface.
   - **local.** If you apply the rule set to an interface as **local**, the rule set filters packets destined for the system itself.

▶ Try it Define a firewall rule set

For our simple example, the natural inclination is to simply create a rule to deny all inbound traffic (that is, from any source network to any destination network) on the Internet-facing interface. The problem with this approach is that outbound connections will not complete properly, because the response packets required to complete these connections will be denied as well.

To circumvent this issue, we explicitly allow only these response packets, as shown in the following example. This rule can be interpreted as, “accept packets from established connections only” (where “established connections” include responses to new connections). Because the final (implicit) rule in the rule set is **deny all**, this rule set will deny all other traffic on the interface destination (that is, **in**, **out**, or **local**) to which it is applied.

```
vyatta@R1# set firewall name ALLOW_ESTABLISHED
```

```
[edit]
vyatta@R1# set firewall name ALLOW_ESTABLISHED rule 10
[edit]
vyatta@R1# set firewall name ALLOW_ESTABLISHED rule 10 action accept
[edit]
vyatta@R1# set firewall name ALLOW_ESTABLISHED rule 10 state established
enable
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

▶ Try it

Apply the rule set to an interface

Now that we have the rule set, we apply it as **in** and **local** on the Internet-facing interface (eth0 in our example) so that connections can only be established from these locations to the Internet.

```
vyatta@R1# set interfaces ethernet eth0 firewall in name ALLOW_ESTABLISHED
[edit]
vyatta@R1# set interfaces ethernet eth0 firewall local name
ALLOW_ESTABLISHED
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

Let's view the firewall rule set we created:

```
vyatta@R1# show firewall
name ALLOW_ESTABLISHED {
    rule 10 {
        action accept
        state {
            established enable
        }
    }
}

[edit]
vyatta@R1#
```

Now let's see this rule set applied as a filter to **in** and **local** on interface eth0:

```
vyatta@R1# show interfaces ethernet
ethernet eth0 {
    address 192.0.2.21/24
    firewall {
        in {
            name ALLOW_ESTABLISHED
        }
        local {
            name ALLOW_ESTABLISHED
        }
    }
    hw-id 00:40:63:e2:e4:00
}
ethernet eth1 {
    address 192.168.1.254/24
    hw-id 00:13:46:e6:f6:87
}
```

This completes configuration of a basic Internet Gateway.