VYATTA, INC. | Evaluation guide

Vyatta—Quick Evaluation Guide

A simple step-by-step guide to configuring network services with Vyatta

VYATTA

Open Source Networking

http://www.vyatta.com

Overview

The purpose of this Evaluation Guide is to introduce you to Vyatta secure router, firewall, and VPN and quickly provide you with enough experience to install Vyatta in your network. This guide will walk you through basic configuration of the system, while familiarizing you with the interface and some of the core features of the product. The step-by-step instructions will allow you to experience the features, flexibility, and ease-of-use that make Vyatta an effective and proven alternative to proprietary networking products.

Note: This guide assumes that you have basic knowledge of networking concepts such as Ethernet, LAN/WAN, NAT, etc.

Vyatta offers network and security services in a Debian Linux-based open-source software package that can transform standard x86 hardware into a secure router, firewall, and VPN. You don't need to be a Linux or open-source "guru" to use this solution. While any x86-based desktop or laptop system with a Vyatta LiveCD will provide you with hands-on experience of the configuration example, access to such a system is not mandatory for following the rest of the document.

In this example we will use Vyatta to connect a small office to the Internet. We will use basic static routing with NAT to provide a large private address space behind the public IP address of the site. We will also cover how to configure an Office DHCP server and a firewall, and how to provide simple DMZ services. We assume that connectivity to the WAN is provided by a second Ethernet interface in the system as shown in Figure 1.

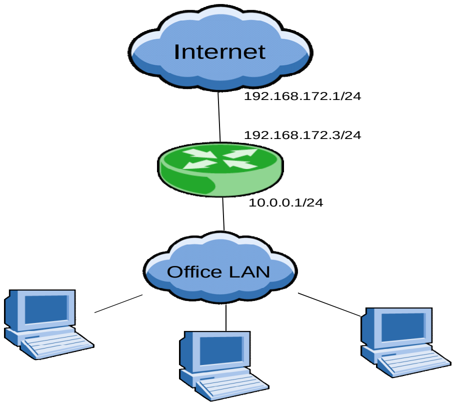


Figure 1: Office Network Example

Booting Up Vyatta

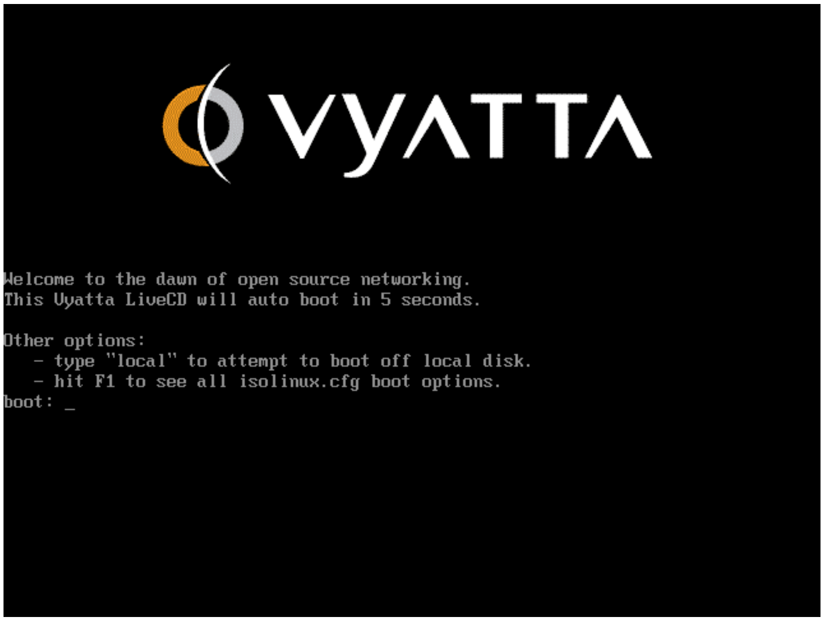
Before you can get a network up and running, you need to boot up the target x86 system with a CD burned with an ISO image of the Vyatta software. The following steps will help you accomplish this.

a) Download the Vyatta software from the Vyatta web site (www.vyatta.com)

b) Burn the ISO image to a CD using your favorite CD burning software (cdrecord or K3B on Linux work well). The CD with the ISO image is a bootable LiveCD containing the Vyatta image.

c) Boot your system with the LiveCD. (You may need to explicitly select the CD-ROM drive as the bootable drive).

Detailed instructions on preparing and booting a system are available in the Vyatta Knowledgebase at: http://www2.vyatta.com/s.nl/sc.7/category.6/ctype.KB/KB.423/.f

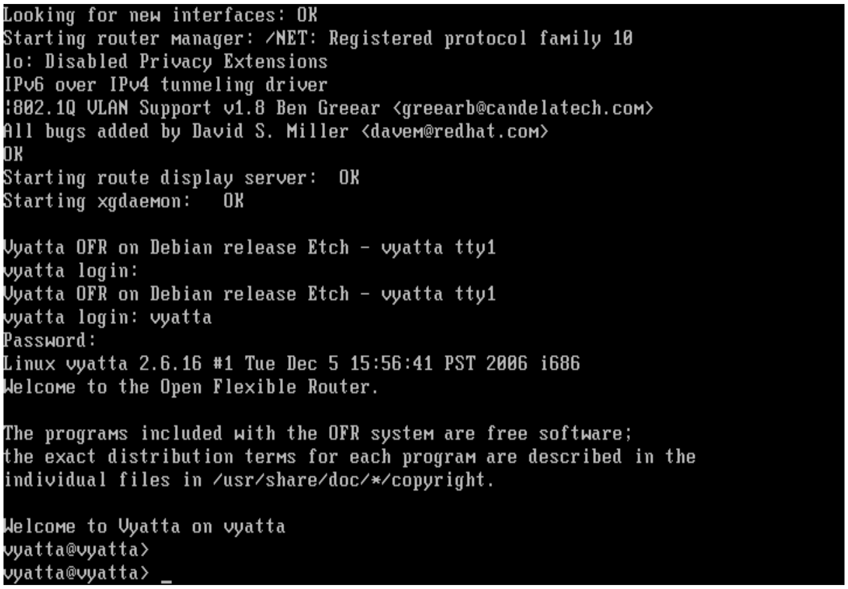


```
Welcome to the dawn of open source networking.
This Vyatta LiveCD will auto boot in 5 seconds.

Other options:
   - type "local" to attempt to boot off local disk.
   - hit F1 to see all isolinux.cfg boot options.
boot: _
```

d) After the system boots up, log in with "**vyatta**" as both the username and password.

After the above steps, you have the target Vyatta system waiting to be configured based on your network parameters.



```
Looking for new interfaces: OK
Starting router manager: /NET: Registered protocol family 10
lo: Disabled Privacy Extensions
IPv6 over IPv4 tunneling driver
802.1Q VLAN Support v1.8 Ben Greear <greearb@candelatech.com>
All bugs added by David S. Miller <davem@redhat.com>
OK
Starting route display server:  OK
Starting xgdaemon:   OK

Vyatta OFR on Debian release Etch - vyatta tty1
vyatta login:
Vyatta OFR on Debian release Etch - vyatta tty1
vyatta login: vyatta
Password:
Linux vyatta 2.6.16 #1 Tue Dec 5 15:56:41 PST 2006 i686
Welcome to the Open Flexible Router.

The programs included with the OFR system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Welcome to Vyatta on vyatta
vyatta@vyatta>
vyatta@vyatta> _
```

Configuring Vyatta

After logging in you will be using the Vyatta shell to enter commands. Similar to the command-line interface on proprietary routers, the Vyatta shell differs substantially from a Linux command-line shell. You can get help with any command by typing "**?**" at any point during the entry of a command. This help feature is useful for exploring the system and viewing the options available for each command.

ENTERING CONFIGURATION MODE

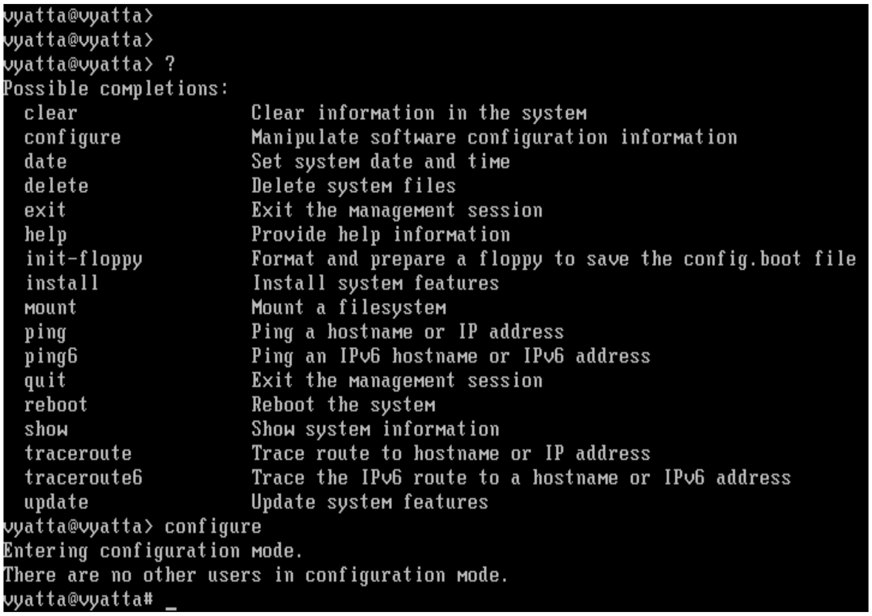
The shell has the concept of an operational mode, where you can view the status of the system, and a configuration mode, where you can configure the system. Enter configuration mode by typing

```
vyatta@vyatta> configure
```

The system puts you in configuration mode with the following message:

```
Entering configuration mode.
There are no other users in configuration mode.
vyatta@vyatta#
```

Notice that the prompt ends with a "**#**" character when in configuration mode, whereas it ends with a "**>**" character in operational mode. In configuration mode, the prompt begins with a location summary, the "**[edit]**" text on the line before the formal prompt. This helps when editing deeply hierarchical configuration structures.



```
vyatta@vyatta>
vyatta@vyatta>
vyatta@vyatta> ?
Possible completions:
  clear                 Clear information in the system
  configure             Manipulate software configuration information
  date                  Set system date and time
  delete                Delete system files
  exit                  Exit the management session
  help                  Provide help information
  init-floppy           Format and prepare a floppy to save the config.boot file
  install               Install system features
  mount                 Mount a filesystem
  ping                  Ping a hostname or IP address
  ping6                 Ping an IPv6 hostname or IPv6 address
  quit                  Exit the management session
  reboot                Reboot the system
  show                  Show system information
  traceroute            Trace route to hostname or IP address
  traceroute6           Trace the IPv6 route to a hostname or IPv6 address
  update                Update system features
vyatta@vyatta> configure
Entering configuration mode.
There are no other users in configuration mode.
vyatta@vyatta# _
```

CONFIGURING INTERFACES

We begin configuring the Vyatta system with the two Ethernet interfaces used for LAN and WAN, respectively. The very basic Ethernet interface configuration includes two parameters: IP address and subnet mask (prefix-length). Here we also configure a text description for each interface to help us easily distinguish between the LAN interface and the WAN interface.

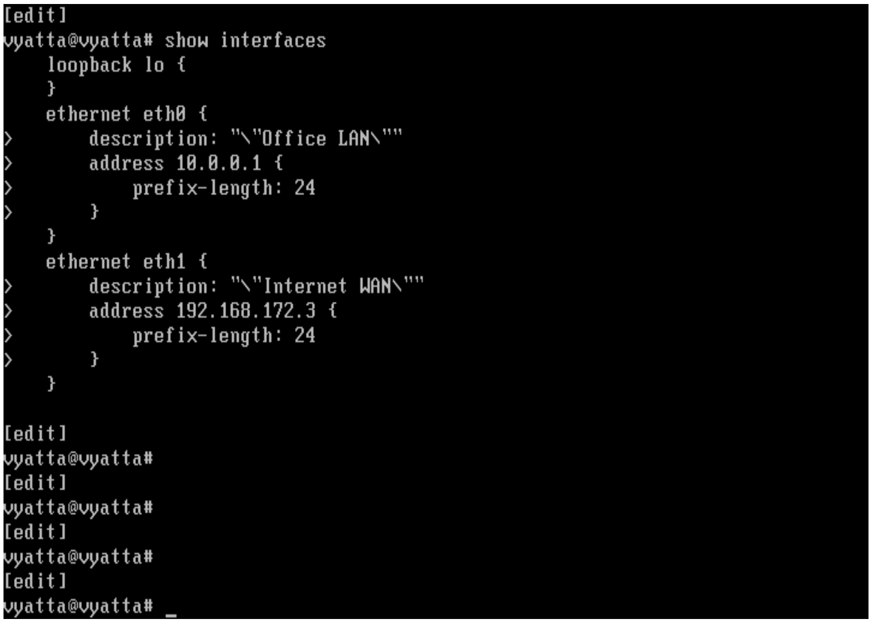
Let’s first configure the Ethernet interface for the office LAN.

```
vyatta@vyatta# set interfaces ethernet eth0 address 10.0.0.1 prefix-length 24
vyatta@vyatta# set interfaces ethernet eth0 description "Office LAN"
```

Next let’s configure the Ethernet interface used to connect to the WAN.

```
vyatta@vyatta# set interfaces ethernet eth1 address 192.168.172.3 prefix-length 24
vyatta@vyatta# set interfaces ethernet eth1 description "Internet WAN"
```

At the end of the above steps, use the ”**show interfaces**” command in configuration mode to check the configuration commands applied to Ethernet eth0 and eth1 interfaces.



```
[edit]
vyatta@vyatta# show interfaces
    loopback lo {
    }
    ethernet eth0 {
>       description: "\"Office LAN\""
>       address 10.0.0.1 {
>           prefix-length: 24
>       }
    }
    ethernet eth1 {
>       description: "\"Internet WAN\""
>       address 192.168.172.3 {
>           prefix-length: 24
>       }
    }

[edit]
vyatta@vyatta#
[edit]
vyatta@vyatta#
[edit]
vyatta@vyatta#
[edit]
vyatta@vyatta# _
```

Note that the ">" in front of any configuration parameter implies that the suggested configuration changes have not been "committed" or applied to the Vyatta system yet. They will take effect only when you issue an explicit "**commit**" command in configuration mode.

Let's configure a few other parameters before committing all the changes.

ROUTING PROTOCOLS

In a more complex scenario, we might configure BGP on the Internet interface or OSPF on the LAN interface. For now, we'll just keep it simple and use a default static route to the Internet.

```
vyatta@vyatta# set protocols static route 0.0.0.0/0 next-hop 192.168.172.1
```

NETWORK ADDRESS TRANSLATION

We're using a private address space inside the office (10.0.0.0/24). We'll need to use NAT to translate the private address space to the public Internet address. The following steps will accomplish this.

Create the configuration node for the NAT rule.

```
vyatta@vyatta# set service nat rule 1
```

Indicate that the rule translates source IP addresses.

```
vyatta@vyatta# set service nat rule 1 type source
```

Set the translation type to masquerade the original source address.

```
vyatta@vyatta# set service nat rule 1 translation-type masquerade
```

Apply the NAT rule on traffic egressing out of Ethernet WAN interface eth1.

```
vyatta@vyatta# set service nat rule 1 outbound-interface eth1
```

Perform NAT on all protocol traffic. ("**all**" is the default option).

```
vyatta@vyatta# set service nat rule 1 protocols all
```

Define the source address or network (in our case 10.0.0.0) to be translated.

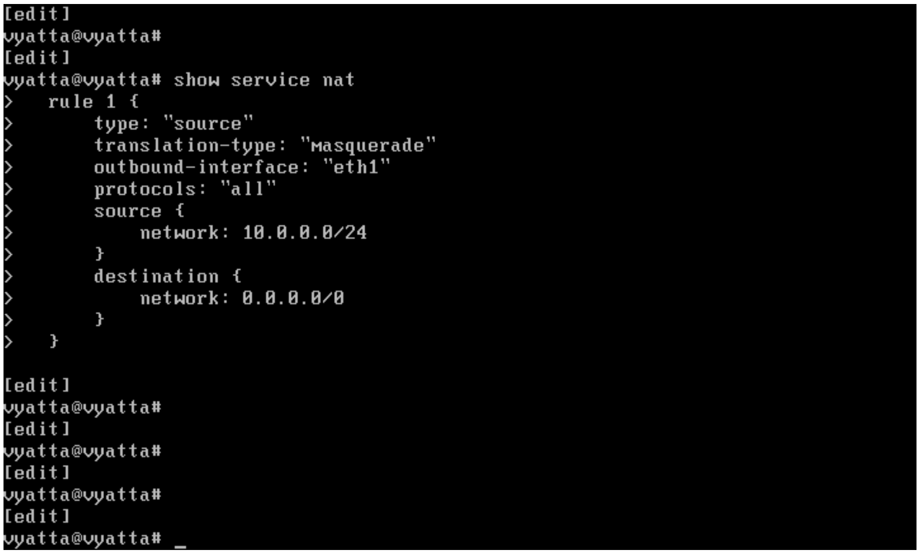
```
vyatta@vyatta# set service nat rule 1 source network 10.0.0.0/24
```

Define the destination network of packets (in our case all networks) translated by this rule.

```
vyatta@vyatta# set service nat rule 1 destination network 0.0.0.0/0
```

This sets up NAT so that all packets exiting the Internet WAN interface eth1, sourced from the 10.0.0.0/24 subnet, will have their source address altered to the address of the eth1 interface. This allows the whole office LAN to share a single IP address on the Internet.

You can see the NAT working configuration waiting to be applied by executing the "**show service nat**" command



```
[edit]
vyatta@vyatta#
[edit]
vyatta@vyatta# show service nat
>    rule 1 {
>        type: "source"
>        translation-type: "masquerade"
>        outbound-interface: "eth1"
>        protocols: "all"
>        source {
>            network: 10.0.0.0/24
>        }
>        destination {
>            network: 0.0.0.0/0
>        }
>    }

[edit]
vyatta@vyatta#
[edit]
vyatta@vyatta#
[edit]
vyatta@vyatta#
[edit]
vyatta@vyatta# _
```

Verifying the Configuration and Committing

Vyatta differs from some of the common closed-source routers in that configuration commands do not take effect as they are typed. Rather, the commands edit a configuration that is then ``committed'' as a single operation. This behavior allows for the configuration to be temporarily inconsistent while it is being edited.

To view the configuration and verify that everything is correct, type

```
vyatta@vyatta# show
```

while in configuration mode. This command displays the complete configuration and allows you to page through it. If you have made a mistake at any point, reenter a "**set**" command with the correct parameters. You can also delete whole sections of the configuration with the "**delete**" command.

When the configuration is correct, type

```
vyatta@vyatta# commit
```

You can then exit back to operational mode by typing

```
vyatta@vyatta# exit
```

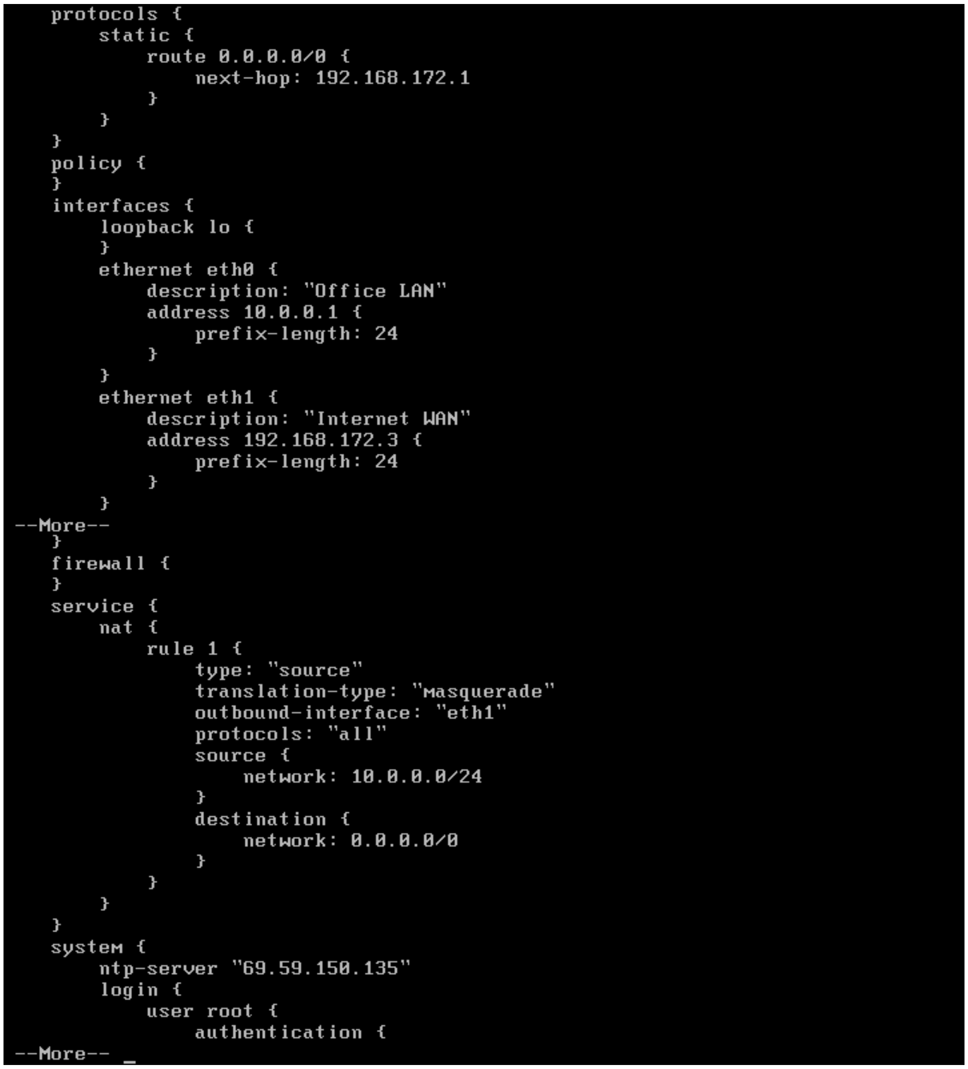
If there is an error in the configuration, the Vyatta system will display a message. You can then correct the error and try to recommit. If you want to throw away the configuration changes before committing the new version, you can type

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

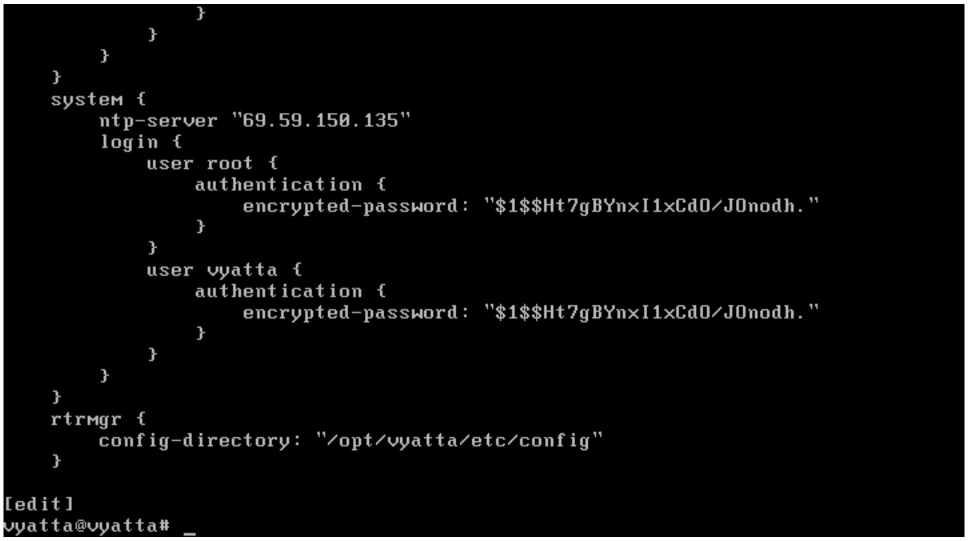
Note that you can enter the configuration mode any time and type the "**show**" command to display the committed active configuration.

Attached below is the output of the "**show**" command at the configuration prompt after we have followed the configuration commands required to configure our small branch network.

```
vyatta@vyatta# show
```



```
    protocols {
        static {
            route 0.0.0.0/0 {
                next-hop: 192.168.172.1
            }
        }
    }
    policy {
    }
    interfaces {
        loopback lo {
        }
        ethernet eth0 {
            description: "Office LAN"
            address 10.0.0.1 {
                prefix-length: 24
            }
        }
        ethernet eth1 {
            description: "Internet WAN"
            address 192.168.172.3 {
                prefix-length: 24
            }
        }
--More--
    }
    firewall {
    }
    service {
        nat {
            rule 1 {
                type: "source"
                translation-type: "masquerade"
                outbound-interface: "eth1"
                protocols: "all"
                source {
                    network: 10.0.0.0/24
                }
                destination {
                    network: 0.0.0.0/0
                }
            }
        }
    }
    system {
        ntp-server "69.59.150.135"
        login {
            user root {
                authentication {
--More-- _
```

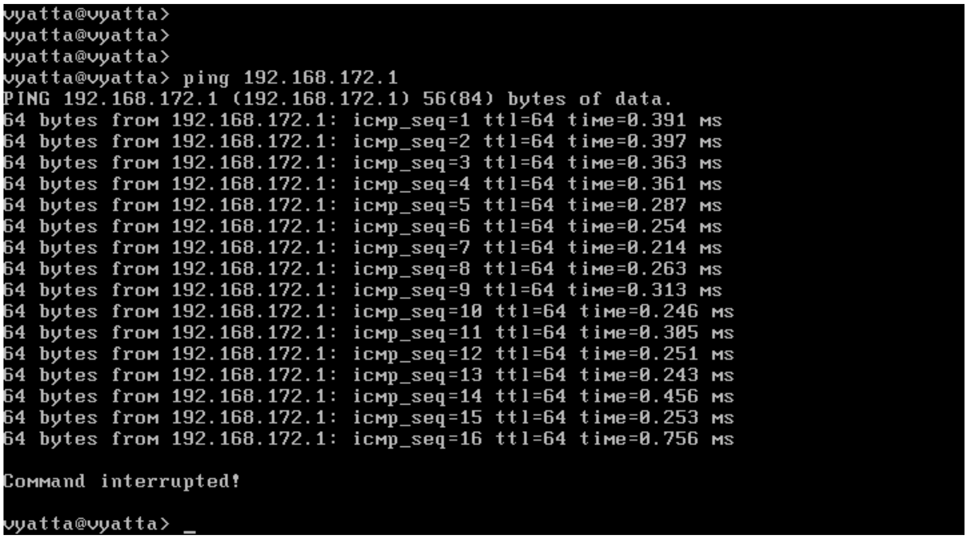


```
                }
            }
        }
    }
    system {
        ntp-server "69.59.150.135"
        login {
            user root {
                authentication {
                    encrypted-password: "$1$$Ht7gBYnxI1xCdO/JOnodh."
                }
            }
            user vyatta {
                authentication {
                    encrypted-password: "$1$$Ht7gBYnxI1xCdO/JOnodh."
                }
            }
        }
    }
    rtrmgr {
        config-directory: "/opt/vyatta/etc/config"
    }

[edit]
vyatta@vyatta# _
```

We validate connectivity to the Internet by issuing a "**ping**" command from the operational mode to the configured gateway address.

```
vyatta@vyatta> ping 192.168.172.1
```



```
vyatta@vyatta>
vyatta@vyatta>
vyatta@vyatta>
vyatta@vyatta> ping 192.168.172.1
PING 192.168.172.1 (192.168.172.1) 56(84) bytes of data.
64 bytes from 192.168.172.1: icmp_seq=1 ttl=64 time=0.391 ms
64 bytes from 192.168.172.1: icmp_seq=2 ttl=64 time=0.397 ms
64 bytes from 192.168.172.1: icmp_seq=3 ttl=64 time=0.363 ms
64 bytes from 192.168.172.1: icmp_seq=4 ttl=64 time=0.361 ms
64 bytes from 192.168.172.1: icmp_seq=5 ttl=64 time=0.287 ms
64 bytes from 192.168.172.1: icmp_seq=6 ttl=64 time=0.254 ms
64 bytes from 192.168.172.1: icmp_seq=7 ttl=64 time=0.214 ms
64 bytes from 192.168.172.1: icmp_seq=8 ttl=64 time=0.263 ms
64 bytes from 192.168.172.1: icmp_seq=9 ttl=64 time=0.313 ms
64 bytes from 192.168.172.1: icmp_seq=10 ttl=64 time=0.246 ms
64 bytes from 192.168.172.1: icmp_seq=11 ttl=64 time=0.305 ms
64 bytes from 192.168.172.1: icmp_seq=12 ttl=64 time=0.251 ms
64 bytes from 192.168.172.1: icmp_seq=13 ttl=64 time=0.243 ms
64 bytes from 192.168.172.1: icmp_seq=14 ttl=64 time=0.456 ms
64 bytes from 192.168.172.1: icmp_seq=15 ttl=64 time=0.253 ms
64 bytes from 192.168.172.1: icmp_seq=16 ttl=64 time=0.756 ms

Command interrupted!

vyatta@vyatta> _
```

At this point the network is operational with connectivity between the LAN in the office and the Internet.

Setting Up Services

We have demonstrated how to set up a small office network with NAT running on an x86-based system in a very short time. Let's now go ahead and configure our router to provide some additional services like DHCP and firewall.

DHCP

In our simple scenario, we'll set up the router to act as the DHCP server for the office LAN. This is not required; you could just as easily set up a server to perform the same task. In our case, we'll serve clients with addresses in the range 10.0.0.50 to 10.0.0.150. This leaves us with some static addresses at the lower end of our subnet numbering for infrastructure devices like printers. You can adjust the size of the DHCP range according to the needs of your particular office. The "**default-router**" option should be set with the IP address of the appropriate default router, which in our case is just this router. The "**interface**" option specifies on which interface the router will answer DHCP requests. We don't want to serve DHCP addresses to the Internet, so we'll just specify the Ethernet eth0 interface.

Follow the steps below to configure the Vyatta system as a DHCP server.

Set up the DHCP server named "OfficeLAN" and configure the pool of IP addresses.

```
        vyatta@vyatta# set service dhcp-server name OfficeLAN start 10.0.0.50 stop
10.0.0.150
```

Set up the subnet mask for the network.

```
        vyatta@vyatta# set service dhcp-server name OfficeLAN network-mask 24
```

Optionally specify the DNS name server available to the clients.

```
        vyatta@vyatta# set service dhcp-server name OfficeLAN dns-server 192.168.172.200
```

Set the Ethernet eth0 address of the Vyatta system as default router for DHCP clients on the 10.0.0.0 network.

```
        vyatta@vyatta# set service dhcp-server name OfficeLAN default-router 10.0.0.1
```

Each DHCP address pool is associated with an interface on the router. Associate the IP address pool with the Ethernet eth0 interface in our case.

```
        vyatta@vyatta# set service dhcp-server name OfficeLAN interface eth0
```

Set up the client domain name to be configured.

```
        vyatta@vyatta# set service dhcp-server name OfficeLAN domain-name mycompany.com
```

FIREWALL

In this section we demonstrate how to set up a simple firewall rule to prevent Telnet access from the Internet into the Vyatta router.

Let's first enable Telnet service on the Vyatta system so that local servers on the LAN can still continue to Telnet into the router for administrative purposes. Enabling Telnet service is accomplished by a simple command.

```
        vyatta@vyatta# set service telnet
```

Now we will set up firewall rules to disallow Telnet access from the Internet.

First, let's create a firewall instance called "FWTELNET" and a set of rules associated with it.

```
        vyatta@vyatta# set firewall name FWTELNET
```

Next, let's create a set of rules for the firewall instance FWTELNET.

```
        vyatta@vyatta# set firewall name FWTELNET rule 1
```

Specify the policy associated with this rule. In our case we want to "**reject**" Telnet packets.

```
vyatta@vyatta# set firewall name FWTELNET rule 1 action reject
```

Specify the protocol for the packets on which this rule will be applied. Telnet uses "**tcp**" as the layer 4 protocol.

```
vyatta@vyatta# set firewall name FWTELNET rule 1 protocol tcp
```

Specify the source network or address of the packets. We will use 0/0 to indicate any network.

```
vyatta@vyatta# set firewall name FWTELNET rule 1 source network 0.0.0.0/0
```

Specify the destination address, network or port. We will use the port name "**telnet**".

```
vyatta@vyatta# set firewall name FWTELNET rule 1 destination port-name telnet
```

At this point, we have configured a firewall rule to block Telnet traffic. However, note that there is always an implicit firewall rule that drops all packets on an interface after the explicitly configured firewall rules are applied. If we define only "**rule 1**" for the firewall instance FWTELNET and apply this on an interface, all packets will be dropped for the reason mentioned above. So, let's define a second rule that accepts all other packets.

```
vyatta@vyatta# set firewall name FWTELNET rule 2

vyatta@vyatta# set firewall name FWTELNET rule 2 action accept

vyatta@vyatta# set firewall name FWTELNET rule 2 protocol all

vyatta@vyatta# set firewall name FWTELNET rule 2 source network 0.0.0.0/0

vyatta@vyatta# set firewall name FWTELNET rule 2 destination network 0.0.0.0/0
```

The firewall rules are applied to packets in the order in numerical order. Rule 1 filters out all Telnet packets. Rule 2 allows all other packets.

Now that we have configured the required firewall rules, let's bind this firewall instance FWTELNET to the Ethernet eth1 interface (which is our interface to the Internet). A firewall does not have any effect on the traffic traversing the network until it is applied to a physical or virtual interface on the Vyatta system. You can apply a firewall rule set for packets coming "**in,**" packets going "**out,**" or packets destined as "**local**" for an interface. In our case, we will apply FWTELNET to all local packets to filter out Telnet attempts on the Vyatta system. You can also filter out Telnet packets destined for hosts in our local LAN.

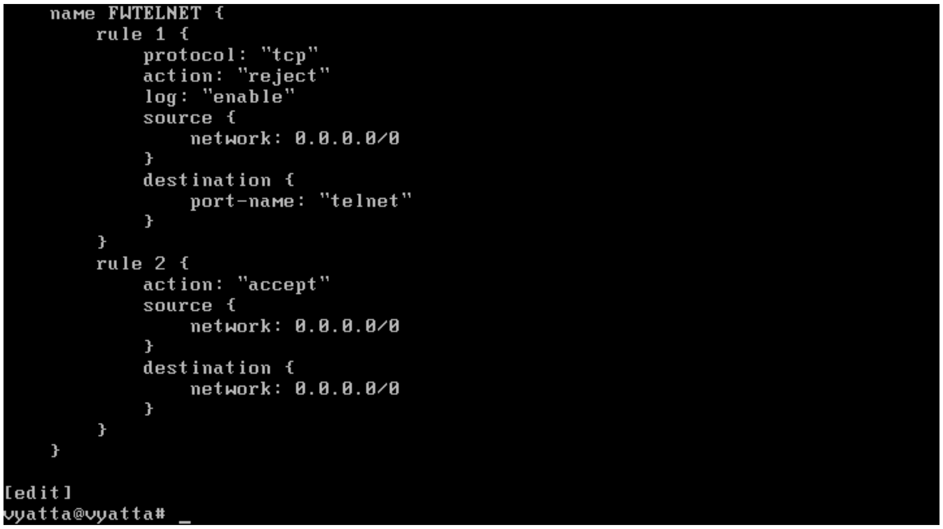
```
vyatta@vyatta# set interfaces ethernet eth1 firewall local name FWTELNET
```

As before, we commit our firewall configuration using the "**commit**" command.

```
vyatta@vyatta# commit
```

The firewall configuration can be seen by the "**show firewall**" command at the configuration prompt.

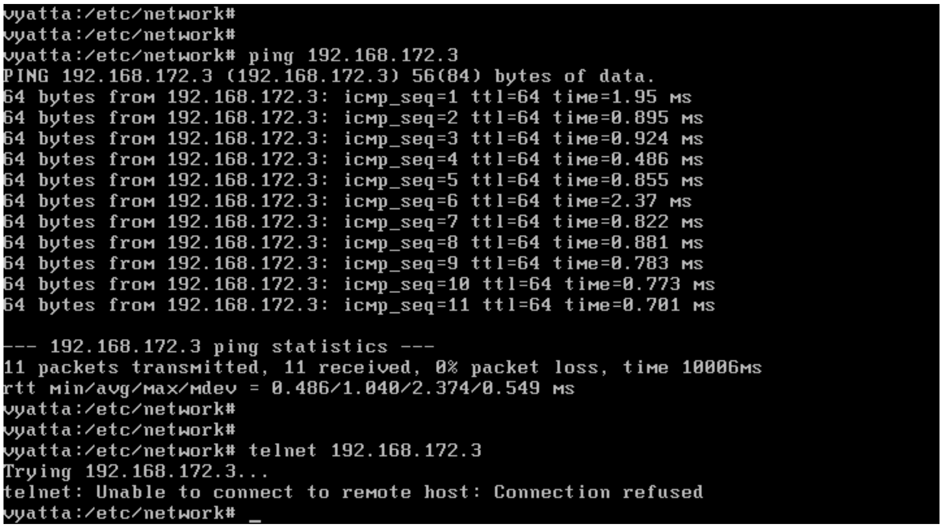
```
vyatta@vyatta# show firewall
```



```
    name FWTELNET {
        rule 1 {
            protocol: "tcp"
            action: "reject"
            log: "enable"
            source {
                network: 0.0.0.0/0
            }
            destination {
                port-name: "telnet"
            }
        }
        rule 2 {
            action: "accept"
            source {
                network: 0.0.0.0/0
            }
            destination {
                network: 0.0.0.0/0
            }
        }
    }

[edit]
vyatta@vyatta# _
```

We verify connectivity to the Vyatta system is still up from a host on the 192.168.172.0 network while Telnet access is disallowed.



```
vyatta:/etc/network#
vyatta:/etc/network#
vyatta:/etc/network# ping 192.168.172.3
PING 192.168.172.3 (192.168.172.3) 56(84) bytes of data.
64 bytes from 192.168.172.3: icmp_seq=1 ttl=64 time=1.95 ms
64 bytes from 192.168.172.3: icmp_seq=2 ttl=64 time=0.895 ms
64 bytes from 192.168.172.3: icmp_seq=3 ttl=64 time=0.924 ms
64 bytes from 192.168.172.3: icmp_seq=4 ttl=64 time=0.486 ms
64 bytes from 192.168.172.3: icmp_seq=5 ttl=64 time=0.855 ms
64 bytes from 192.168.172.3: icmp_seq=6 ttl=64 time=2.37 ms
64 bytes from 192.168.172.3: icmp_seq=7 ttl=64 time=0.822 ms
64 bytes from 192.168.172.3: icmp_seq=8 ttl=64 time=0.881 ms
64 bytes from 192.168.172.3: icmp_seq=9 ttl=64 time=0.783 ms
64 bytes from 192.168.172.3: icmp_seq=10 ttl=64 time=0.773 ms
64 bytes from 192.168.172.3: icmp_seq=11 ttl=64 time=0.701 ms

--- 192.168.172.3 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10006ms
rtt min/avg/max/mdev = 0.486/1.040/2.374/0.549 ms
vyatta:/etc/network#
vyatta:/etc/network#
vyatta:/etc/network# telnet 192.168.172.3
Trying 192.168.172.3...
telnet: Unable to connect to remote host: Connection refused
vyatta:/etc/network# _
```

DMZ SERVICE

Typically, when provisioning an office network, one may want to keep the private office network on a LAN segment separate from a network used to offer e-mail, HTTP, FTP and other services to clients on the Internet. In our example with one Ethernet interface providing LAN connectivity for all office devices, we will configure a simple NAT rule such that the Vyatta router, acting as a proxy HTTP server, permits HTTP traffic to a server in the Office LAN network.

Create the configuration node for the NAT rule.

```
vyatta@vyatta# set service nat rule 2
```

Indicate that the rule translates destination IP address.

```
        vyatta@vyatta# set service nat rule 2 type destination
```

Set the translation type to static.

```
        vyatta@vyatta# set service nat rule 2 translation-type static
```

Apply the NAT rule on traffic ingressing into the Ethernet WAN interface eth1.

```
        vyatta@vyatta# set service nat rule 2 inbound-interface eth1
```

Perform NAT on TCP protocol traffic destined for the HTTP port on 192.168.172.3.

```
        vyatta@vyatta# set service nat rule 2 protocol tcp
        vyatta@vyatta# set service nat rule 2 destination address 192.168.172.3
        vyatta@vyatta# set service nat rule 2 destination port-name http
```

Define the source address or network (``any'' in our case).

```
        vyatta@vyatta# set service nat rule 2 source network 0.0.0.0/0
```

Define the destination or forwarding address of packets (HTTP server).

```
        vyatta@vyatta# set service nat rule 2 inside-address address 10.0.0.30
```

This sets up NAT so that all http packets entering the Vyatta system on the Ethernet eth1 interface are forwarded to the HTTP server 10.0.0.30.

Now the Vyatta system has been configured to provide routing, DHCP, NAT and firewall services for a small office network. You can use this configuration as a building block for your own network. For additional help on the Vyatta CLI or for more configuration examples, please consult with the Vyatta OFR Command Reference or Vyatta OFR Configuration Guide available under Vyatta documentation at http://www.vyatta.com/documentation/

Conclusion

Using Vyatta software and an x86-based PC or server, in less than thirty minutes you can build an enterprise-class router/firewall for a fraction of the cost of an equivalent closed-source, proprietary router. Vyatta is simple to configure and operate and can scale from the small, basic configurations that we have built in these examples, up through larger, more complex network infrastructures, including VPN features.

Vyatta offers excellent value in the following environments:

- SMB / Enterprise / Campus LAN Segmentation
- SMB / Enterprise / Campus LAN routing
- ISP / Enterprise Edge Routing to T3 speeds
- ISP CPE for Managed Services
- Hosting / Datacenter routing

FEEDBACK

Have comments on this paper? Send them to feedback@vyatta.com