

## Scenario: V114 Configuration on Vyatta

This section steps you through initial system configuration tasks. These are tasks that are required for almost any scenario in which you might use the V114 on the Vyatta system. These include:

### **Vyatta:**

- Logging On
- Entering Configuration Mode
- Configuring Interfaces
- Configuring Access to DNS Server
- Configuring DHCP Server
- Configuring NAT
- Configuring Firewall to allow SIP/RTP traffic to the V114 from internet.

### **V114:**

- LED on the V114
- Accessibility to the Positron-GUI
- Configuring Interface

*Figure 1 shows a network diagram of the V114 configuration scenario.*

**Note:** This whitepaper assumes we are using a standard vyatta live CD thus the V114's PCI driver is not compiled into the vyatta source code.

## Logging On

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

### **Procedure:**

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 2.6.30-1-586-vyatta #2 SMP Tue Sep 15 18:34:05 CEST 2009 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.
vyatta@vyatta:~$
```

## Entering Configuration Mode

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



### Procedure:

Enter configuration mode by entering **configure**.

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

**Note:** The command prompt changes to mark the move from operational mode (“:~\$”) to configuration mode (“#”).

## 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 be configured.

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, while ‘eth1’ is the Ethernet Interface facing the Local area network.

The loopback interface, which is a software interface automatically, created on startup is preconfigured to IP address ‘127.0.0.1/8’. The loopback interface will 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.

### □ Procedure for configuring an Internet-facing Ethernet interface

Assuming that ISP provided the dynamic IP address, the interface ‘eth0’ is configured as below.

```
vyatta@vyatta # set interfaces ethernet eth0 address dhcp
[edit]
vyatta@vyatta # commit
[edit]
vyatta@vyatta#
```

### □ Procedure for configuring the Office LAN-facing Ethernet interface

```
vyatta@vyatta # set interfaces ethernet eth1 address 192.168.5.1/24
[edit]
vyatta@vyatta # commit
[edit]
vyatta@vyatta#
```

To view the configuration we use the show command:

```
vyatta@vyatta# show interfaces
    ethernet eth0 {
        address dhcp
        hw-id 08:00:27:d0:ea:b3
    }
```



```
    ethernet eth1 {
        address 192.168.5.1/24
        hw-id 08:00:27:d0:ea:03
    }
    loopback lo {
    }
[edit]
vyatta@vyatta#
```

**Note:** The default gateway for the internet facing Ethernet interface is obtained from the dhcp IP address provided by the ISP.

## 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.

### □ Procedure for Specifying a DNS server

In our example, the DNS server IP Address is provided by the ISP, i.e. '24.200.243.189'. Add the DNS server using the '`set system name-server`' command.

```
vyatta@vyatta# set system name-server 24.200.243.189
[edit]
vyatta@vyatta# commit
[edit]
vyatta@vyatta#
```

## 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`').

### □ Procedure for setting up DHCP Server

For the DHCP server, define an address pool from '192.168.5.100 to 192.168.5.199' 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@vyatta# set service dhcp-server shared-network-name ETH1_POOL subnet
192.168.5.0/24 start 192.168.5.100 stop 192.168.5.199
[edit]
vyatta@vyatta# set service dhcp-server shared-network-name ETH1_POOL subnet
192.168.5.0/24 default-router 192.168.5.1
[edit]
vyatta@vyatta# set service dhcp-server shared-network-name ETH1_POOL subnet
192.168.5.0/24 dns-server 24.200.243.189
[edit]
vyatta@vyatta# commit
[edit]
vyatta@vyatta#
```



To view the configuration we use the below command:

```
vyatta@vyatta# show service dhcp-server
  shared-network-name ETH1_POOL {
    subnet 192.168.5.0/24 {
      dns-server 24.200.243.189
      default-router 192.168.5.1
      start 192.168.5.100 {
        stop 192.168.5.199
      }
    }
  }
[edit]
vyatta@vyatta#
```

## Configuring NAT

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

### □ Procedure to define a NAT rule

Define a rule that allows traffic from network ‘192.168.5.0/24’ to proceed to the Internet through interface ‘eth0’, and translates any internal addresses to eth0’s IP address. (This is called “masquerade” translation.)

```
vyatta@vyatta# set service nat rule 1 source address 192.168.5.0/24
[edit]
vyatta@vyatta# set service nat rule 1 outbound-interface eth0
[edit]
vyatta@vyatta# set service nat rule 1 type masquerade
[edit]
vyatta@vyatta# commit
[edit]
vyatta@vyatta#
```

We would also need to configure the nat rules so that the remote users can register their SIP Phones from internet to the V114. Nat rules 2, 3 are used for this purpose.

```
vyatta@vyatta# set service nat rule 2 description SIP-V114
[edit]
vyatta@vyatta# set service nat rule 2 destination port 5060-5065
[edit]
vyatta@vyatta# set service nat rule 2 inbound-interface eth0
[edit]
vyatta@vyatta# set service nat rule 2 inside-address address 192.168.5.10
[edit]
vyatta@vyatta# set service nat rule 2 inside-address port 5060-5065
[edit]
vyatta@vyatta# set service nat rule 2 protocol udp
[edit]
vyatta@vyatta# set service nat rule 2 type destination
[edit]
vyatta@vyatta# commit
```



```
[edit]
vyatta@vyatta#

vyatta@vyatta# set service nat rule 3 description RTP-V114
[edit]
vyatta@vyatta# set service nat rule 3 destination port 10001-20000
[edit]
vyatta@vyatta# set service nat rule 3 inbound-interface eth0
[edit]
vyatta@vyatta# set service nat rule 3 inside-address address 192.168.5.10
[edit]
vyatta@vyatta# set service nat rule 3 inside-address port 10001-20000
[edit]
vyatta@vyatta# set service nat rule 3 protocol udp
[edit]
vyatta@vyatta# set service nat rule 3 type destination
[edit]
vyatta@vyatta# commit
[edit]
vyatta@vyatta#
```

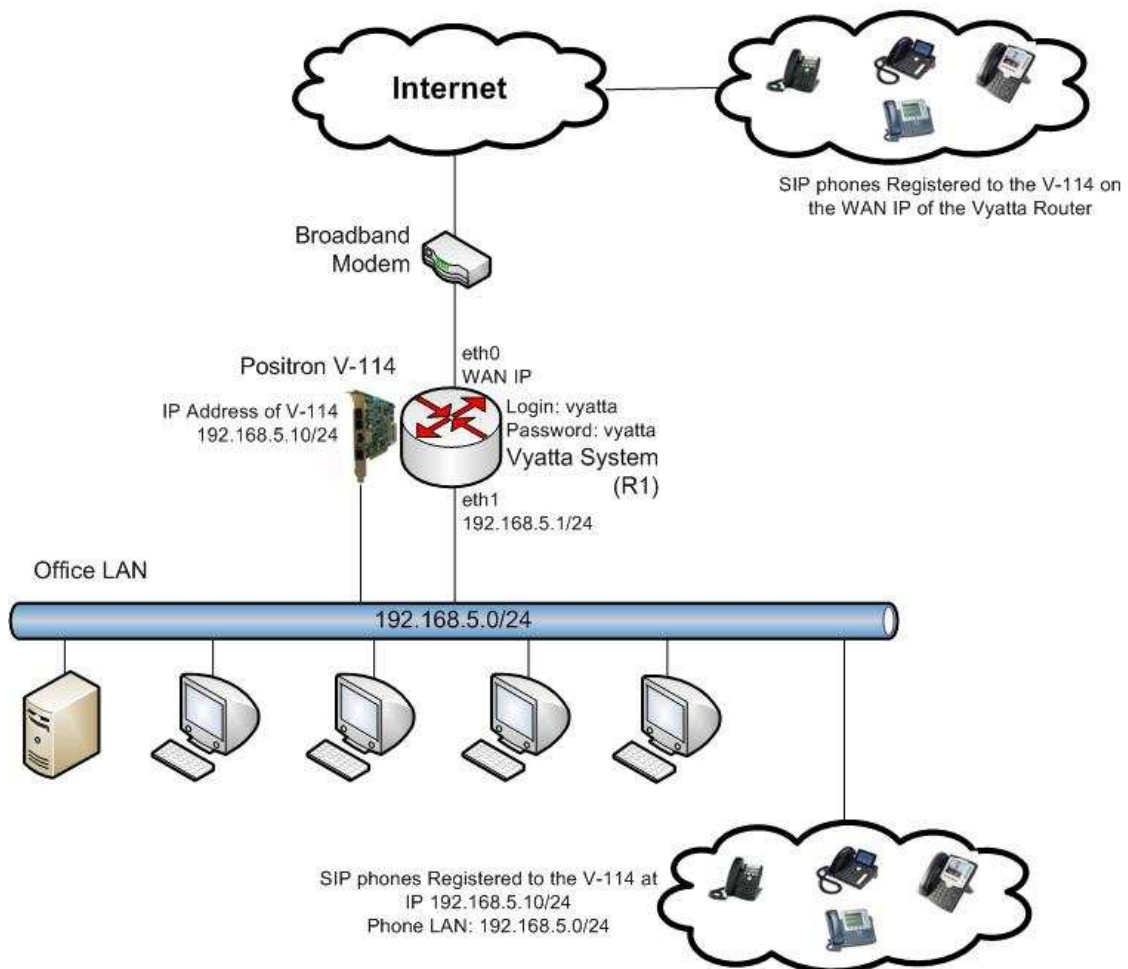
To view the configuration we use the below command:

```
vyatta@vyatta# show service nat
rule 1 {
    outbound-interface eth0
    source {
        address 192.168.5.0/24
    }
    type masquerade
}
rule 2 {
    description SIP-V114
    destination {
        port 5060-5065
    }
    inbound-interface eth0
    inside-address {
        address 192.168.5.10
        port 5060-5065
    }
    protocol udp
    type destination
}
rule 3 {
    description RTP-V114
    destination {
        port 10001-20000
    }
    inbound-interface eth0
    inside-address {
        address 192.168.5.10
        port 10001-20000
    }
    protocol udp
    type destination
}
[edit]
vyatta@vyatta#
```

## Network Design

The Network design used for this whitepaper is as shown in Figure 1. A Positron V-114 is installed in the Vyatta Router in one of the PCI Slots.

### Positron V-114 in Vyatta Router without the PCI Driver Installed



**Figure 1: Network Design**

## Configuring Firewall

As it is shipped, the Vyatta System does not restrict traffic flow through it, unless a firewall rule is applied. The firewall functionality provides packet filtering which enables great 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 that is 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 implicitly denied, because there is an implicit ‘**deny all**’ rule at the end of any rule set.

In general, to configure a firewall on an interface:

- You define a number of named firewall rule sets, each of which contains one or more firewall rules.
- You apply the each of the named rule sets to an interface as a filter. You can apply one named rule set as 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.

#### □ Procedure to define a firewall rule set

Consider a simple example; the natural inclination is to simply create a rule to deny all inbound traffic (i.e. 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 response packets required to complete these connections will be denied as well. To circumvent this issue we must explicitly allow only these response packets as shown in the following example. This can be interpreted as “*accept packets from established connections only*” (where “established connections” includes 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 (i.e. **in**, **out**, or **local**) that it is applied to.

```
vyatta@vyatta# set firewall name ETH0IN
[edit]
vyatta@vyatta# set firewall name ETH0IN rule 10
[edit]
vyatta@vyatta# set firewall name ETH0IN rule 10 action accept
[edit]
vyatta@vyatta# set firewall name ETH0IN rule 10 state established enable
[edit]
vyatta@vyatta# set firewall name ETH0IN rule 20 action accept
[edit]
vyatta@vyatta# set firewall name ETH0IN rule 20 protocol udp
[edit]
vyatta@vyatta# set firewall name ETH0IN rule 20 source address 0.0.0.0/0
[edit]
vyatta@vyatta# set firewall name ETH0IN rule 20 source port 5060-5065
[edit]
vyatta@vyatta# set firewall name ETH0IN rule 20 destination address
192.168.5.10
[edit]
vyatta@vyatta# set firewall name ETH0IN rule 20 state new enable
[edit]
vyatta@vyatta# set firewall name ETH0IN rule 20 state established enable
[edit]
vyatta@vyatta# set firewall name ETH0IN rule 20 state related enable
[edit]
vyatta@vyatta# set firewall name ETH0IN rule 30 action accept
[edit]
vyatta@vyatta# set firewall name ETH0IN rule 30 protocol udp
[edit]
```



```
vyatta@vyatta# set firewall name ETH0IN rule 30 source address 0.0.0.0/0
[edit]
vyatta@vyatta# set firewall name ETH0IN rule 30 source port 10001-20000
[edit]
vyatta@vyatta# set firewall name ETH0IN rule 30 destination address
192.168.5.10
[edit]
vyatta@vyatta# set firewall name ETH0IN rule 30 state new enable
[edit]
vyatta@vyatta# set firewall name ETH0IN rule 30 state established enable
[edit]
vyatta@vyatta# set firewall name ETH0IN rule 30 state related enable
[edit]
vyatta@vyatta# commit
[edit]
vyatta@vyatta#
```

#### □ Procedure for applying the rule set to an interface

Now that we have the rule set, we need to 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@vyatta# set interfaces ethernet eth0 firewall in name ETH0IN
[edit]
vyatta@vyatta# set interfaces ethernet eth0 firewall local name ETH0IN
[edit]
vyatta@vyatta# commit
[edit]
vyatta@vyatta#
```

To view the firewall rule set we created:

```
vyatta@vyatta# show firewall
name ETH0IN {
    rule 10 {
        action accept
        state {
            established enable
        }
    }
    rule 20 {
        action accept
        destination {
            address 192.168.5.10
        }
        protocol udp
        source {
            address 0.0.0.0/0
            port 5060-5065
        }
        state {
            established enable
            new enable
            related enable
        }
    }
    rule 30 {
        action accept
    }
}
```



```
        destination {
            address 192.168.5.10
        }
        protocol udp
        source {
            address 0.0.0.0/0
            port 10001-20000
        }
        state {
            established enable
            new enable
            related enable
        }
    }
}
[edit]
vyatta@vyatta#
```

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

```
vyatta@vyatta# show interfaces ethernet
    ethernet eth0 {
        address dhcp
        firewall {
            in {
                name ETH0IN
            }
            local {
                name ETH0IN
            }
        }
        hw-id 08:00:27:d0:ea:b3
    }
    ethernet eth1 {
        address 192.168.5.1/24
        hw-id 08:00:27:d0:ea:03
    }
}
```

This completes configuration of a basic Internet Gateway.

## LED on the V114

We need to make sure the green LED on the V114 is Solid green which would mean the V114 does not have any software/hardware issues.

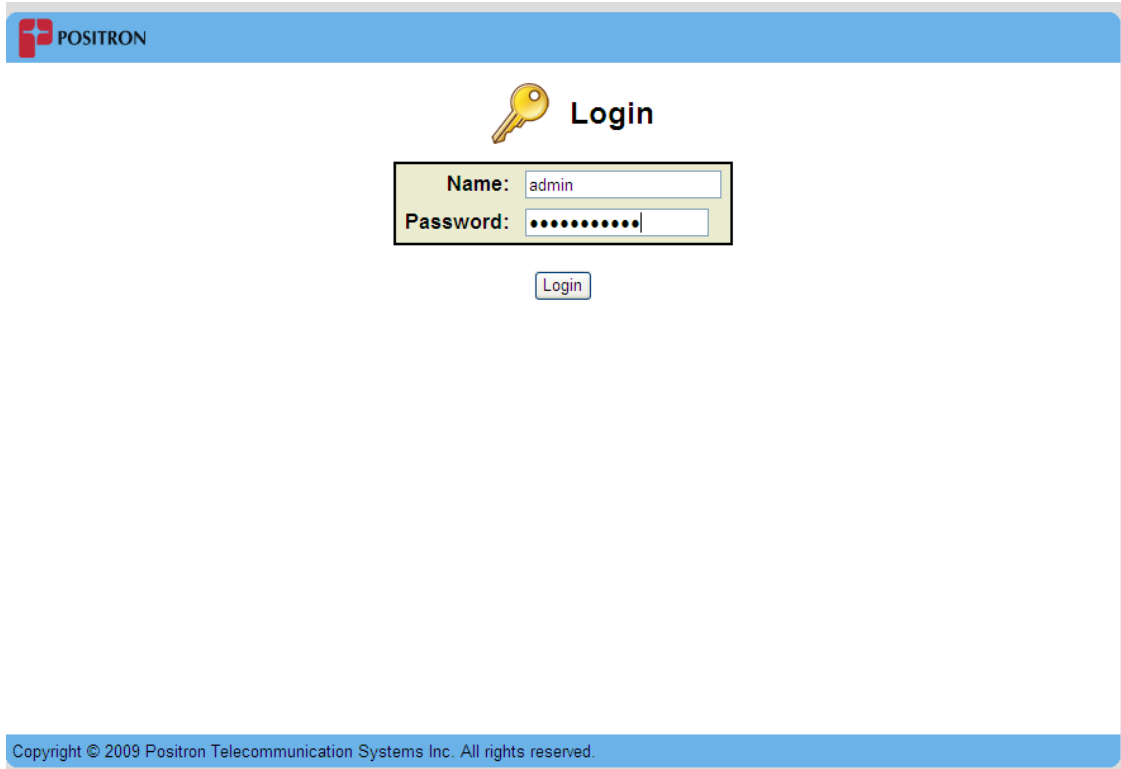
## Accessibility to the Positron-GUI

It is also important to connect a cable from the Ethernet port on the V114 to a PC/Laptop. The V114 has the default IP address as '192.168.1.2/24'. Hence to access the V114, we will assign the IP address of '192.168.1.100/24' to the PC/Laptop connected to the other end of the cable. Once the IP Address is configured, we would check the following.

1. Check to see the link is connected on the PC as 100 MBPS Full duplex connection.
2. Ensure we have successful ping responses to the V114. i.e, a successful "ping 192.168.1.2".



If the ping responses are successful, we can open browser of our choice and type in the IP address of the V114 to access the Positron GUI as shown in Figure 2.



**Figure 2: Positron GUI**

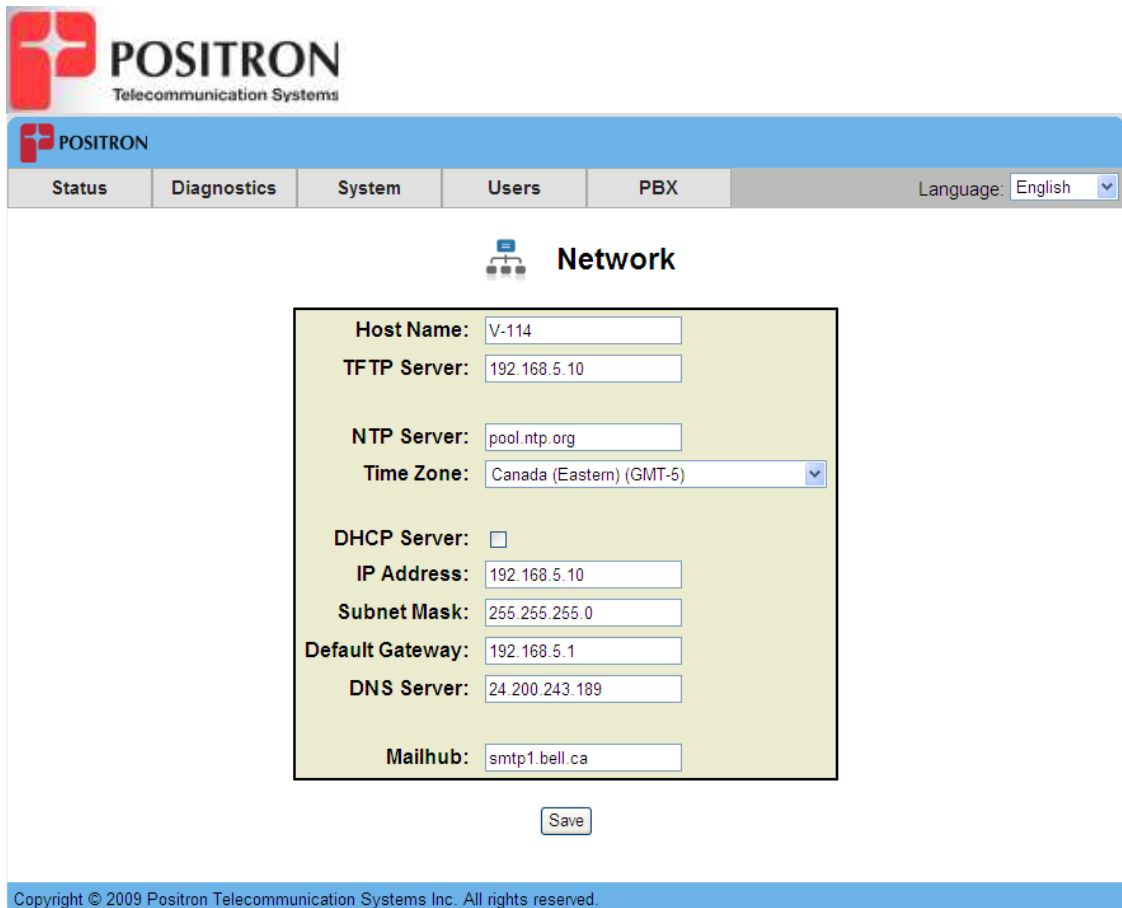
The default username/password for the Positron GUI is admin/password.

If the ping responses are not successful, please check the ip address assigned to the PC/Laptop and if the cable is properly connected.

## Configuring Interface

We configure the IP address of the V114 as 192.168.5.10/24 as shown in the network design (Figure 1) using the Positron GUI – Figure 3.

Please click on the System → Network to configure the Network interface on the V114.



**Figure 3: Network Configuration**

Once we saved the network configuration, we can plug the V114 to the Ethernet port to the LAN switch using an Ethernet cable. We would also change the IP on the PC from '192.168.1.100/24' to its original IP address.

Now you can ping the '192.168.5.10' from any PC or Server in the Office LAN. This also implies you can register all the SIP phones to the V114 over the LAN.

The firewall rules are programmed to accept any SIP traffic on the 'eth0' (WAN) ethernet interface on the vyatta to be redirected to the V114 IP address.

Thus, SIP phones can also be registered to the V114 from the internet.