Vyatta VC5 - Simple Firewall and NAT Rules

Adrian F. Dimcev www.carbonwind.net
contact@carbonwind.net

- 1. Introduction
- 2. Overview
- 3. Firewalling Vyatta itself
  - 3.1 Allow DNS name resolution for Vyatta itself
  - 3.2 Allow NTP for Vyatta itself
  - 3.3 Allow HTTP and HTTPS for Vyatta itself
  - 3.4 Dynamic IP address from DHCP on an interface
  - 3.5 Allow SSH to Vyatta Itself
  - 3.6 Allow ping from Vyatta itself
  - 3.7 Allow TFTP from Vyatta itself to an internal TFTP server
  - 3.8 Allow FTP from Vyatta itself to an internal FTP server
  - 3.9 Allow SCP from Vyatta itself to an internal SCP server and from an internal SCP client to Vyatta itself
  - 3.10 Allow Radius traffic from Vyatta itself to an internal Radius server
  - 3.11 Vyatta as PPTP VPN Server: VPN traffic destined to Vyatta itself
  - 3.12 Vyatta as L2TP/IPsec VPN Server: VPN traffic destined to Vyatta itself
  - 3.13 Vyatta as an IPsec tunnel mode VPN gateway: s2s traffic destined to Vyatta itself
  - 3.14 GRE over IPsec: Traffic destined to Vyatta itself
  - 3.15 Allow OSPF traffic to Vyatta itself
  - 3.16 Allow RIP traffic to Vyatta itself
  - 3.17 Vyatta as DHCP server
  - 3.18 Vyatta as DHCP relay server
  - 3.19 DNS Forwarding on Vyatta
  - 3.20 Vyatta as a web proxy
- 4. Traffic through Vyatta
  - 4.1 Allow FTP through Vyatta
  - 4.2 Allow TFTP through Vyatta
  - 4.3 Allow web traffic through Vyatta
  - 4.4 Allow DNS through Vyatta
  - 4.5 Allow Ping through Vyatta
  - 4.6 Allow PPTP through Vyatta
  - 4.7 Allow L2TP/IPsec through Vyatta
  - 4.8 Vyatta as L2TP/IPsec or PPTP VPN Server: Filter VPN clients' traffic
  - 4.9 Vyatta as an IPsec tunnel mode VPN gateway: s2s traffic between the local and remote subnets and vice-versa
  - 4.10 Vyatta as an IPsec tunnel mode VPN gateway: Excluding from the NAT process traffic destined
    to the remote subnet(s)
  - 4.11 Vyatta as web proxy + Vyatta as IPsec tunnel mode VPN gateway
  - 4.12 GRE over IPsec: Traffic between local and remote subnets on the tunnel interface(tun1)
  - 4.13 GRE over IPsec: Traffic between local and remote subnets on the internal interface(eth1)
  - 4.14 GRE over IPsec: In and out firewall instances on the Internet facing interface(eth0)
- 5. Publish servers with Vyatta
  - 5.1 Publish a web(HTTP) server
  - 5.2 Publish a web(HTTP) server on an alternate port
  - 5.3 Publish a FTP server
  - 5.4 Publish a FTP server on an alternate port
  - 5.5 Publish a SMTP server

- **5.6 Publish a PPTP VPN server**
- **5.7 Publish a L2TP/IPsec or "pure" IPsec VPN server behind Vyatta**
- **6. Small home router behavior**
- **7. Quickly display firewall rules and view firewall statistics**

1. Introduction

Since I've noticed that configuring Vyatta's firewall is a popular topic, I've decided to write this article.
I was not sure if to put it in a blog post, or on the main site, as it is my current understanding that in the future the firewall on Vyatta and the way firewall rules are configured might get some updates, making the bellow lines to need some updates. I actually hope that will be so and the firewall on Vyatta would be more user friendly.

Personal I'm not a big fan on how the firewall can be currently configured on Vyatta VC 5.0.2. Simple put, it's just too easy to get it wrong, or to not obtain the most secure configuration due to some details. And the underlying iptables are currently underused.

I will try to cover some common scenarios(but there are many possible common scenarios), firewalling Vyatta itself or traffic through Vyatta. Over the time I hope to add more configuration examples.

I will not use the Web GUI, as in its current form I don't feel it brings too much in configuring Vyatta's firewall, and it would be difficult for me to picture every step needed to configure a firewall rule. Rather, the Web GUI would be more suited for some videos tutorials.

2. Overview

Before we begin let's talk about some things, that may be useful before considering configuring the firewall:

- in the bellow lines I will use the Vyatta VC5 5.0.2 version.

- before you proceed make sure you read Vyatta's documentation. The publicly available product documentation can be currently found on Vyatta.org, the documentation section. And if the firewall configuration is important for you and you are unsure if you got it right, then it might be time to get professional support.

- firewall is off by default.

- you can configure the firewall from both the CLI and the web GUI, but the GUI right now for the firewall(and not only) may be basically a GUI "emulating" the CLI, still may help you a little bit if you are not familiar with Vyatta's CLI.

- there are no wizards or so to help you configure the firewall rules for basic access or firewall Vyatta itself(basic services enabled on Vyatta), or some firewall templates to start working with.

- stateful inspection is off by default, so you need to use the state parameter on your firewall rules. Without the state parameter on your rules, besides the advantages you get with the stateful inspection, for example, some junk packets, may go un-NAT-ed through Vyatta in case you configured say a masquerade or source NAT rule. For more aboute the state parameter, you may like to read this.

NEW: the packet starts a new connection(like SYN segments for TCP connections).
RELATED:the packet starts a new connection while this connection is associated with an existing connection(say
the FTP data channel)or maybe be an ICMP error packet.
ESTABLISHED: the packet is part from a connection already established.

INVALID: the packet is not associated with any known connections.

- you cannot work with firewall objects or firewall zones.

- you apply firewall instances as **in**, **out** and **local**. See **Figure1**.

- you cannot filter traffic sourced from the Vyatta itself, currently the **local** instance only applies to trafic destined to Vyatta itself on the interface you've applied that firewall instance.

- conntrack-tcp-loose is enabled by default, see **Figure2**, meaning, say "lonely" ACKs are allowed through(for example one can proble for open ports on Vyatta itself like so, although a local firewall instance with "stateful inspection firewall rules" was configured).
So I'm going to disable it bellow every time TCP is involved.

- SYN cookies are off by default.

- when you create a firewall rule and apply as needed a firewall instance, the rest of the traffic is implicitly denied for that firewall instance(normal firewall behavior) as there is a "deny rule" number 1025. See **Figure3**.

- any L7 "intelligence", say FTP, TFTP, etc., is "loaded" by default, and cannot be modified from Vyatta's CLI.

- currently you cannot configure time-based firewall rules from Vyatta's CLI.

- you can configure a basic anti-spoofing mechanism with firewall rules(say using the source parameter on your firewall rules).

- Unicast Reverse Path Forwarding is not available(not from Vyatta's CLI).

- you currently cannot configure from the CLI restrictions for the management(SSH, Web GUI) without the firewall. This means that, for example, if you enable SSH, Vyatta will listen for inbound SSH connections including on the GRE tunnels' IP addresses or on the PPP interface if you enable the remote access VPN server on Vyatta(either PPTP or L2TP/IPsec).

- if you have multiple interfaces, you may need to carefully apply the **in**, **out** or **local** firewall instances on all these interfaces. As for example, a **local** firewall instance on an interface(say eth0) does not apply only to inbound traffic destined to the router itself for that interface, it may apply to "all" inbound traffic destined to the router itself received on that interface(you can reach like so through eth0 **local** firewall instance the IP address of interface eth1 or the IP address of the tunnel interface tun1).

- if you have only two interfaces on Vyatta, you may gain little if you apply both **in** and **out** firewall instances on both interfaces, as you may "get away" with only **in** firewall instances on both interfaces and obtain pretty much the same effect(this may vary based on your scenario though).

- try not to "mix" the firewall rule set, for example when you create a firewall rule set, don't use on it rules for traffic destined to both the Vyatta itself and non-destined to Vyatta itself, and then apply this firewall rule set as both a **local** and **in** firewall instance on an interface. Instead create two firewall rule sets, each for every firewall instance. This will help you keep things clean and logical and prevent any unexpected results.

- some(see this document) may instruct you to open through the firewall for example to allow HTTP, from a client TCP source ports 1024+(more exactly 1024-65535) to a server destination TCP port 80. This is normal

behavior as the client will connect from source port 1024+, however you cannot specify both multiple source and destination ports on Vyatta's firewall.

- I've tried to create a form of protection against SYN flooding through Vyatta(from hosts behind an interface of Vyatta to hosts behind another interface of the same Vyatta, to prevent one behind Vyatta to SYN flood an external web server for example or and external host to SYN flood a web server behind Vyatta) with the recent count and time features but I got stuck with a 20 count per 60 seconds, otherwise I would get an error when committing. I wanted something like drop more than 600 new TCP connections from a host in 60 seconds(I suppose one may try instead 20 new TCP connections say in 2 seconds) -I know it's not pretty as Vyatta will not attempt to SYN proxy, so if one can spoof another host(as it just have to make its spoofed SYN segments reach Vyatta), it may be able to make Vyatta deny access for the real legitimate host-. If you want more details about the recent match, you may like to read this.

- take advantage of the fact that Vyatta can run in a virtualized environment, and so you can do tests and configuration tuning with ease prior of the deployment phase. I found one thing kinda annoying though, for example I saved a "clean" config to a TFTP server prior of adding any firewall rules, and say I start to add firewall rules, and at one point I want to quickly revert to my "clean state"(no firewall rules) using the **load** command to load the clean saved config. Well, it did not go according to plan, as the loaded config is attempted to commit and error from **Figure4** appears, which apparently won't go way until I remove the firewall instances of the interfaces(delete them + commit the changes), which in the end makes pretty useless my plan, as after deleting the firewall instances from the interfaces(delete them + commit the changes) I'm one simple command away for deleting the entire firewall configuration.

- don't forget to monitor your rules to see if they work as expected(even if you applied a rule as the logic(at least your logic -:)) may dictate, it does not mean it will filter traffic, see the dhcp relay example from bellow).

- for convenience when I will configure the firewall bellow, I will not specify the **commit** command, so don't forget to commit your configuration. And if you want your config to be persistent through a reboot don't forget to save it with the **save** command.

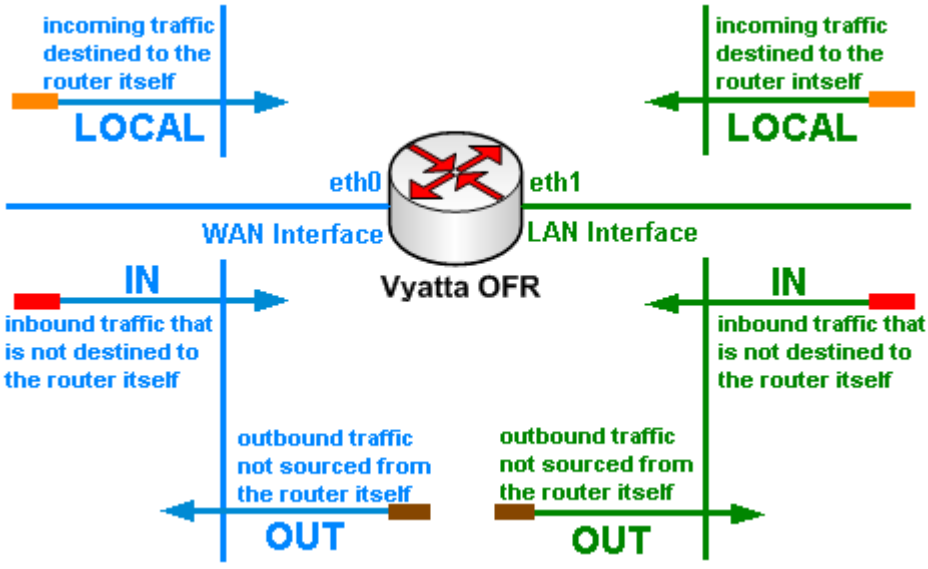

Figure1: Firewall Instances

```
vyatta@vyatta# show firewall -all
 broadcast-ping disable
 conntrack-tcp-loose enable
 ip-src-route disable
 log-martians enable
```

Figure2: Firewall Defaults

```
vyatta@vyatta:~$ show firewall eth0local

Active on (eth0,LOCAL)

State Codes: E - Established, I - Invalid, N - New, R - Related

rule  action  source              destination         proto  state
----  ------  ------              -----------         -----  -----
20    ACCEPT  0.0.0.0/0           0.0.0.0/0           tcp    E,N,R
                                  dst ports: 22
1025  DROP    0.0.0.0/0           0.0.0.0/0           all    any

vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$ show firewall eth0local statistics

Active on (eth0,LOCAL)

rule  packets   bytes     action  source              destination
----  -------   -----     ------  ------              -----------
20    122       8532      ACCEPT  0.0.0.0/0           0.0.0.0/0
1025  1999      87988     DROP    0.0.0.0/0           0.0.0.0/0

vyatta@vyatta:~$
```

Figure3: Rule 1025

```
vyatta@vyatta# load tftp://192.168.40.56/config1
################################################################################ 100.0%
Loading config file /opt/vyatta/etc/config/config.boot.5115...
Firewall config error: Cannot delete rule set "eth0in" (still in use)
Commit failed
Load failed (commit failed)
[edit]
```

Figure4: Load Config Failed

3. Firewalling Vyatta itself

There a lot of possible scenarios here. First, there are a couple of returning connections for services that Vyatta router needs, as DNS for name resolution, HTTP(and maybe HTTPS) for updates, NTP for time synchronization with the configured NTP server.
Let's take a look, see **Figure5** for the network diagram.

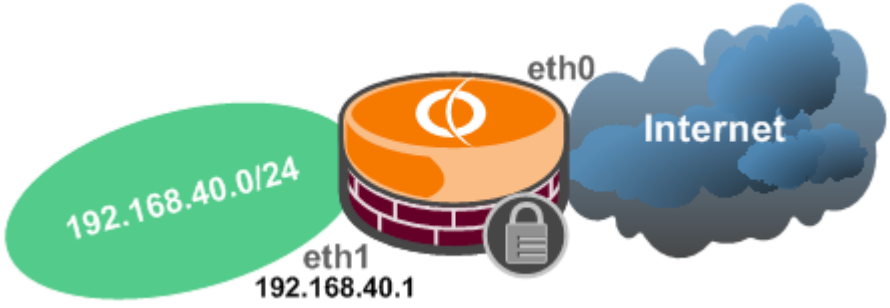

Figure5: Network Diagram

3.1 Allow DNS name resolution for Vyatta itself

- Maybe you obtain the IP address on your Internet facing interface through DHPC, so you do not know the IP addresses of the ISP DNS servers, as their IP addresses might change.

```
set firewall name eth0local rule 10 action accept
set firewall name eth0local rule 10 protocol udp
set firewall name eth0local rule 10 source port 53
set firewall name eth0local rule 10 state established enable
set firewall name eth0local rule 10 state related enable

set interfaces ethernet eth0 firewall local name eth0local
```

- Or maybe you have a static IP address on your Internet facing interface, and you know the IP addresses of the ISP DNS servers. Bellow, we have assumed that the public IP address on eth0 interface is 192.168.22.240 and the IP address of the ISP server is 192.168.22.1.

```
set firewall name eth0local rule 10 action accept
set firewall name eth0local rule 10 protocol udp
set firewall name eth0local rule 10 source port 53
set firewall name eth0local rule 10 source address 192.168.22.1
set firewall name eth0local rule 10 destination address 192.168.22.240
set firewall name eth0local rule 10 state established enable
set firewall name eth0local rule 10 state related enable

set interfaces ethernet eth0 firewall local name eth0local
```

- Or you may use internal DNS server(s) and you've configured on Vyatta this DNS server(s) as name server(s)(note that in addition to the rule bellow you need to allow DNS through Vyatta from the internal DNS, search this article for that).

```
set firewall name eth1local rule 10 action accept
set firewall name eth1local rule 10 protocol udp
set firewall name eth1local rule 10 source address 192.168.40.2
set firewall name eth1local rule 10 source port 53
set firewall name eth1local rule 10 destination address 192.168.40.1
set firewall name eth1local rule 10 state established enable
set firewall name eth1local rule 10 state related enable

set interfaces ethernet eth1 firewall local name eth1local
```

3.2 Allow NTP for Vyatta itself

Say you've configured an NTP server on Vyatta. By default there is an external NTP server configured. For this configured NTP server, we can add on the Internet facing interface a firewall rule allowing returning NTP traffic(NTP server's replies) like (bellow, we have assumed that the public IP address on eth0 interface is 192.168.22.240 and the IP address of the NTP server is 69.59.150.135):

```
set firewall name eth0local rule 10 action accept
set firewall name eth0local rule 10 protocol udp
set firewall name eth0local rule 10 source address 69.59.150.135
set firewall name eth0local rule 10 source port 123
set firewall name eth0local rule 10 destination address 192.168.22.240
set firewall name eth0local rule 10 state established enable
```

```
set firewall name eth0local rule 10 state related enable

set interfaces ethernet eth0 firewall local name eth0local
```

3.3 Allow HTTP and HTTPS for Vyatta itself

Say you want to allow for updates or so, HTTP/HTTPS traffic from Vyatta itself(destined to the Internet). So you need to allow returning HTTP/HTTPS traffic to Vyatta itself on that interface(connections initiated by Vyatta itself, bellow, we have assumed that the public IP address on eth0 interface is 192.168.22.240).

```
set firewall conntrack-tcp-loose disable

set firewall name eth0local rule 10 action accept
set firewall name eth0local rule 10 protocol tcp
set firewall name eth0local rule 10 source port 80,443
set firewall name eth0local rule 10 destination address 192.168.22.240
set firewall name eth0local rule 10 state established enable
set firewall name eth0local rule 10 state related enable

set interfaces ethernet eth0 firewall local name eth0local
```

3.4 Dynamic IP address from DHCP on an interface

Say, the Internet facing interface(eth0) obtains its IP configuration from a DHCP server.
Basically we need to allow the DHCP server's reply messages.

```
set firewall name eth0local rule 10 action accept
set firewall name eth0local rule 10 protocol udp
set firewall name eth0local rule 10 destination port 68
set firewall name eth0local rule 10 source port 67
set firewall name eth0local rule 10 state established enable
set firewall name eth0local rule 10 state related enable

set interfaces ethernet eth0 firewall local name eth0local
```

3.5 Allow SSH to Vyatta Itself

You enable for management the SSH service on Vyatta. As already said, it will listen on every interface. Also you are worried about SSH brute-force attacks. There might be more elegant and efficient solutions for protection against brute-force for SSH outside the CLI, but we can work out something with firewall rules. Also if you allow SSH on the Internet facing interface, it may be a good idea to change the default port for SSH(sometimes security-by-obscurity can help a little, at least when one probes for the default SSH port).

Say you want to add a rule allowing SSH on the eth0 interface:
- rule 10 drops new connections from a specific source IP address if from that source IP address more than 5 new connections were initiated within 60 seconds. For example, if I send five SYN segments from my IP address, Vyatta will SYN-ACK only the first five, and then deny the sixth. If after the sixth I back off for about 60 seconds, I will be allowed to initiate new connections again. Note that one, if it may spoof its IP address can easily block access to legitimate SSH clients with rule 10 in place.
- rule 20 allows the SSH traffic to Vyatta itself

Note: We did not specify within rule 20 the destination IP address of the Vyatta machine, and if so, you may be able to SSH-in from the Internet destining packets to Vyatta's eth1 IP address(well, private IP address space should not be routed over the Internet), even if you have a masquerade NAT rule on interface eth0(also if the interfaces eth1 was firewalled with a local firewall instance, this firewall instance will be applied only to packets arriving on the eth1 interface, and in this case the packets are arriving on interface eth0).

Also, if you have multiple interfaces, say also eth2, eth3, etc, connecting internal subnets, you may be able to SSH-in from the Internet destining packets to Vyatta's eth2/eth3 IP addresses.

```
set firewall conntrack-tcp-loose disable

set firewall name eth0local rule 10 action drop
set firewall name eth0local rule 10 protocol tcp
set firewall name eth0local rule 10 destination port 22
set firewall name eth0local rule 10 state new enable
set firewall name eth0local rule 10 recent count 5
set firewall name eth0local rule 10 recent time 60

set firewall name eth0local rule 20 action accept
set firewall name eth0local rule 20 protocol tcp
set firewall name eth0local rule 20 destination port 22
set firewall name eth0local rule 20 state new enable
set firewall name eth0local rule 20 state established enable
set firewall name eth0local rule 20 state related enable

set interfaces ethernet eth0 firewall local name eth0local
```

To take care of the above note, we have a couple of possibilities.

- First, if you have a dynamic IP address on that interface and you use dynamic DNS or so, you cannot specify the destination IP address.
The most simple thing to do, is to add drop rules for destination IP addresses from private IP address space(I know this is not quite pretty). Note that in the bellow example we assume you have private IP addresses on the rest of the interfaces(like eth1). Also note that traffic will be dropped only when received on the eth0 interface. If you SSH-in over a VPN remote access connection(say PPTP), your SSH traffic for the IP address of eth1 will not be dropped.

```
set firewall conntrack-tcp-loose disable

set firewall name eth0local rule 10 action drop
set firewall name eth0local rule 10 protocol tcp
set firewall name eth0local rule 10 destination port 22
set firewall name eth0local rule 10 state new enable
set firewall name eth0local rule 10 recent count 5
set firewall name eth0local rule 10 recent time 60

set firewall name eth0local rule 15 action drop
set firewall name eth0local rule 15 protocol tcp
set firewall name eth0local rule 15 destination port 22
set firewall name eth0local rule 15 destination address 192.168.0.0/16
set firewall name eth0local rule 15 state new enable
set firewall name eth0local rule 15 state established enable
set firewall name eth0local rule 15 state related enable

set firewall name eth0local rule 16 action drop
set firewall name eth0local rule 16 protocol tcp
set firewall name eth0local rule 16 destination port 22
set firewall name eth0local rule 16 destination address 172.16.0.0/12
set firewall name eth0local rule 16 state new enable
```

```
set firewall name eth0local rule 16 state established enable
set firewall name eth0local rule 16 state related enable

set firewall name eth0local rule 17 action drop
set firewall name eth0local rule 17 protocol tcp
set firewall name eth0local rule 17 destination port 22
set firewall name eth0local rule 17 destination address 10.0.0.0/8
set firewall name eth0local rule 17 state new enable
set firewall name eth0local rule 17 state established enable
set firewall name eth0local rule 17 state related enable

set firewall name eth0local rule 20 action accept
set firewall name eth0local rule 20 protocol tcp
set firewall name eth0local rule 20 destination port 22
set firewall name eth0local rule 20 state new enable
set firewall name eth0local rule 20 state established enable
set firewall name eth0local rule 20 state related enable

set interfaces ethernet eth0 firewall local name eth0local
```

- If you have a static IP address on that interface, things are more simple, we can specify the destination IP address. Bellow we assumed that 192.168.22.240 was the public IP address from eth0 interface.

```
set firewall conntrack-tcp-loose disable

set firewall name eth0local rule 10 action drop
set firewall name eth0local rule 10 protocol tcp
set firewall name eth0local rule 10 destination port 22
set firewall name eth0local rule 10 state new enable
set firewall name eth0local rule 10 recent count 5
set firewall name eth0local rule 10 recent time 60

set firewall name eth0local rule 20 action accept
set firewall name eth0local rule 20 protocol tcp
set firewall name eth0local rule 20 destination port 22
set firewall name eth0local rule 20 destination address 192.168.22.240
set firewall name eth0local rule 20 state new enable
set firewall name eth0local rule 20 state established enable
set firewall name eth0local rule 20 state related enable

set interfaces ethernet eth0 firewall local name eth0local
```

We may want to exempt or so certain IP addresses from the 5 new connections in 60 seconds rule. Say we want a certain IP address to be not be allowed to initiate more than 10 new connections in 60 seconds(maybe there is a NAT device somewhere and there are multiple users behind it) -actually we don't necessarily need the exclusion from rule 10, we could just placed rule 11 before rule 10-:

- within rule 10, similar with the previous one, this time we exclude the needed IP address(you can find more about firewall exclusion within Vyatta's publicly available documentation, the Security Reference Guide)
- rule 11 drops new connections from the needed IP address if from that source IP address more than 10 new connections were initiated within 60 seconds.
- rule 20 allows SSH traffic

```
set firewall conntrack-tcp-loose disable

set firewall name eth0local rule 10 action drop
set firewall name eth0local rule 10 protocol tcp
set firewall name eth0local rule 10 destination port 22
set firewall name eth0local rule 10 source address !192.168.22.159
set firewall name eth0local rule 10 state new enable
set firewall name eth0local rule 10 recent count 5
set firewall name eth0local rule 10 recent time 60

set firewall name eth0local rule 11 action drop
set firewall name eth0local rule 11 protocol tcp
set firewall name eth0local rule 11 destination port 22
set firewall name eth0local rule 11 source address 192.168.22.159
set firewall name eth0local rule 11 state new enable
set firewall name eth0local rule 11 recent count 10
set firewall name eth0local rule 11 recent time 60

set firewall name eth0local rule 20 action accept
set firewall name eth0local rule 20 protocol tcp
set firewall name eth0local rule 20 destination port 22
set firewall name eth0local rule 20 destination address 192.168.22.240
set firewall name eth0local rule 20 state new enable
set firewall name eth0local rule 20 state established enable
set firewall name eth0local rule 20 state related enable

set interfaces ethernet eth0 firewall local name eth0local
```

3.6 Allow ping from Vyatta itself

For connectivity tests, you may be able to ping from Vyatta itself.

So we need to allow the ICMP echo-reply messages for established connections(connections for which the router has sent the ICMP echo-request message and is waiting the echo-reply message, bellow, we have assumed that the IP address on eth0 interface is 192.168.22.240).

```
set firewall name extl0cal rule 10 action accept
set firewall name extl0cal rule 10 protocol icmp
set firewall name extl0cal rule 10 destination address 192.168.22.240
set firewall name extl0cal rule 10 icmp type 0
set firewall name extl0cal rule 10 icmp code 0
set firewall name extl0cal rule 10 state established enable
set firewall name extl0cal rule 10 state related enable

set interfaces ethernet eth0 firewall local name eth0local
```

3.7 Allow TFTP from Vyatta itself to an internal TFTP server

You may want to save/load Vyatta's configuration to an internal TFTP server.

TFTP is say, "special". For example, we want to save or load the configuration to the 192.168.40.56 TFTP server(I'm using TFTPD32).
For convenience, I have specified a local ports pool on the TFTP server (44440:44450), thus Tftpd32 assigns the range of ports used for file transfers from that pool.
For a save attempt, first there will be a Write Request from Vyatta to the destination UDP port 69 (TFTP

port) and then there will be a reply from the TFTP server to Vyatta, an Acknowledgement packet, with the source UDP port from the range 44440-44450 and not 69. So the firewall should be aware of that and dynamically allow TFTP traffic.

I will use TFtpd32 as the TFTP server, see **Figure6**.

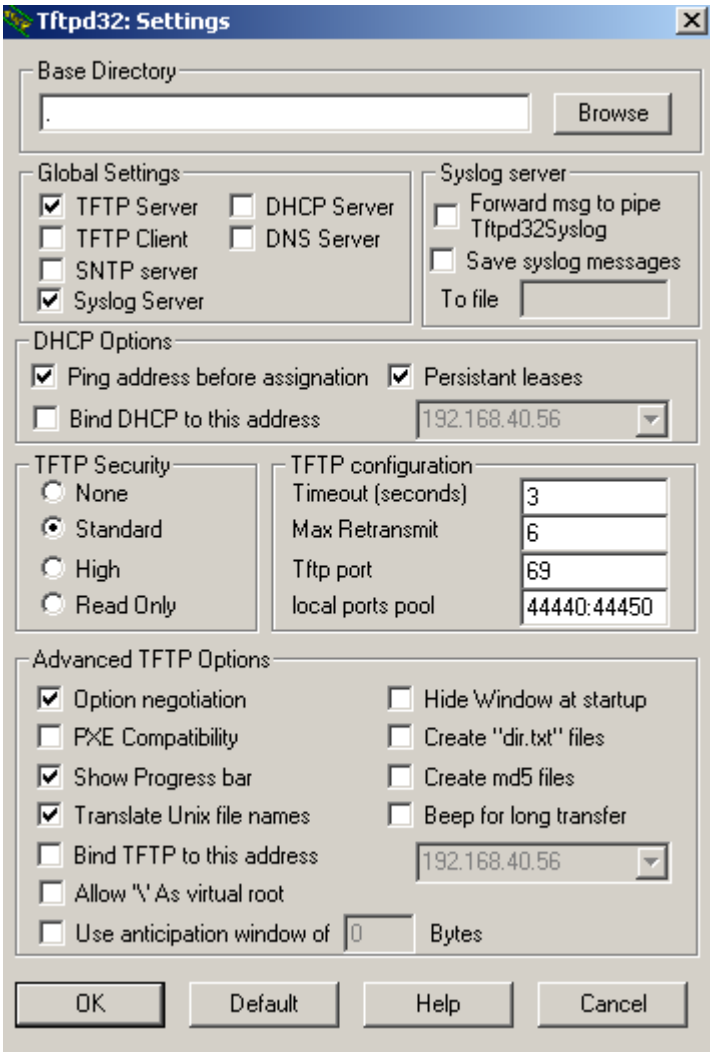

Figure6: TFTP Server settings

Luckily, the needed TFTP module was already loaded, see **Figure7**, thus we don't need to allow new connections from the source port range 44440-44450, just related and established ones.

Figure7: Loaded TFTP modules

The firewall rule, note that we don't need to allow any replies sourced from UDP port 69, as there aren't any(and the initial connection to UDP port 69 on the TFTP server is allowed) -bellow, we have assumed that the IP address on eth1 interface is 192.168.40.1 and the TFTP server's IP address is 192.68.40.56-:

```
set firewall name eth1local rule 10 action accept
set firewall name eth1local rule 10 source address 192.168.40.56
set firewall name eth1local rule 10 destination address 192.168.40.1
set firewall name eth1local rule 10 protocol udp
set firewall name eth1local rule 10 state established enable
set firewall name eth1local rule 10 state related enable
set firewall name eth1local rule 10 source port 44440-44450

set interfaces ethernet eth1 firewall local name eth1local
```

3.8 Allow FTP from Vyatta itself to an internal FTP server

You may want to save/load Vyatta's configuration to an internal FTP server.

FTP is also say, "special". For example, we want to save or load the configuration to the 192.168.40.56 FTP server(I'm using Filezilla FTP Server).
For convenience, I have specified a local ports pool on the FTP server (44440:44450) for Passive FTP, see **Figure8**, thus Filezilla FTP Server assigns the range of ports used for file transfers from that pool for the data channel for Passive FTP.
I've noticed that Vyatta VC 5.0.2 requests EPSV by default.

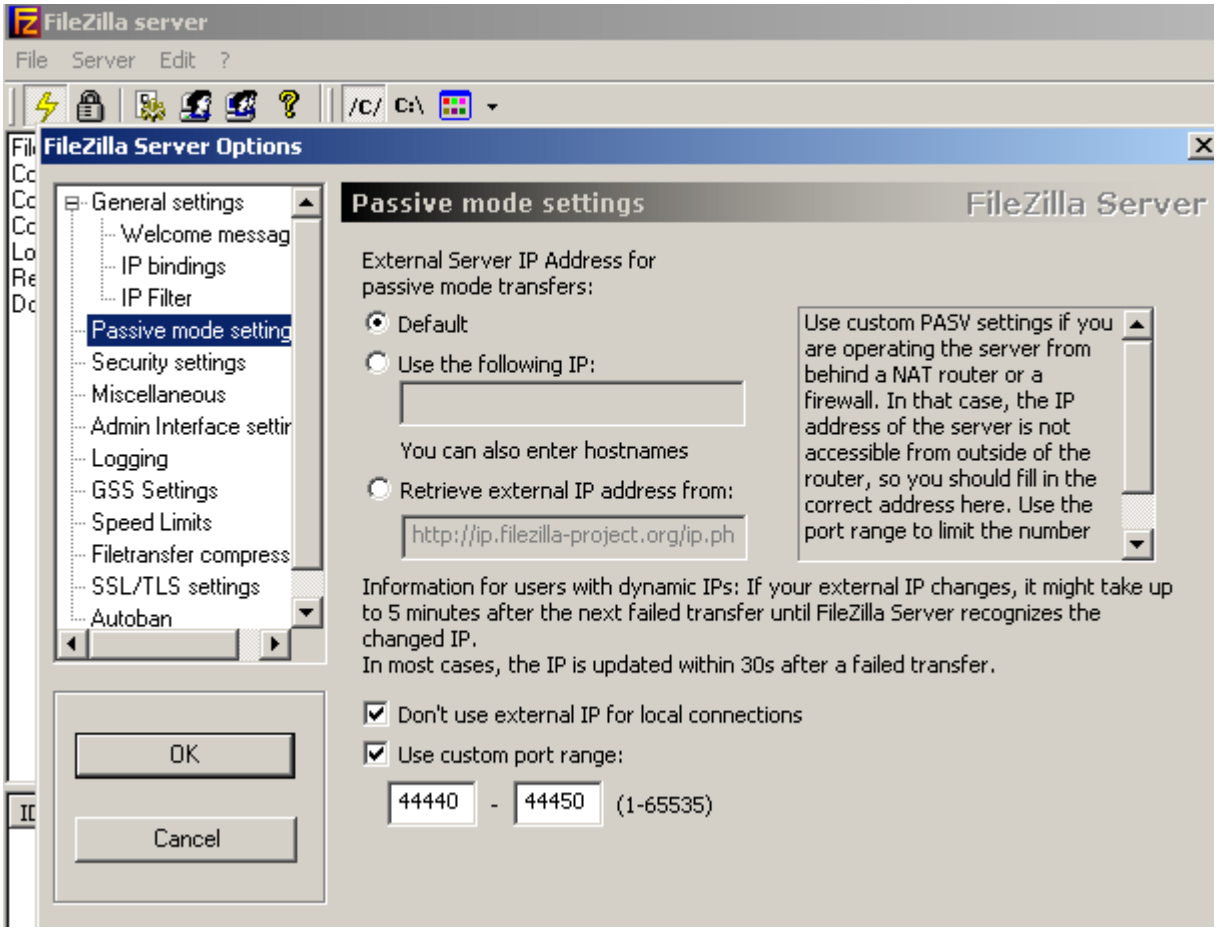

Figure8: Filezilla Passive FTP Settings

The needed FTP modules were already loaded, see **Figure9**, but anyway we don't need to allow new connections from the source port range 44440-44450, just related and established ones.

```
vyatta@vyatta:~$ zcat /proc/config.gz | grep -i ftp
CONFIG_IP_VS_FTP=m
CONFIG_NF_CONNTRACK_FTP=m
CONFIG_NF_CONNTRACK_TFTP=m
CONFIG_NF_NAT_FTP=m
CONFIG_NF_NAT_TFTP=m
vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$ cat /proc/modules | grep -i ftp
nf_nat_tftp 2432 0 - Live 0xd0bca000
nf_nat_ftp 3712 0 - Live 0xd0bc8000
nf_nat 18072 7 iptable_nat,nf_nat_pptp,nf_nat_h323,nf_nat_sip,nf_nat_proto_gre,n
f_nat_tftp,nf_nat_ftp, Live 0xd0be8000
nf_conntrack_tftp 5140 1 nf_nat_tftp, Live 0xd0b8b000
nf_conntrack_ftp 8356 1 nf_nat_ftp, Live 0xd0bc4000
nf_conntrack 61140 15 iptable_nat,xt_NOTRACK,nf_nat_pptp,nf_conntrack_pptp,nf_co
nntrack_proto_gre,nf_nat_h323,nf_conntrack_h323,nf_nat_sip,nf_conntrack_sip,nf_n
at_tftp,nf_nat_ftp,nf_nat,nf_conntrack_ipv4,nf_conntrack_tftp,nf_conntrack_ftp,
Live 0xd0bf8000
vyatta@vyatta:~$
```

Figure9: Loaded FTP modules

The firewall rules(bellow we have assumed that the IP address on eth1 interface is 192.168.40.1 and the FTP server's IP address is 192.68.40.56-):

- rule 10 allows established connections for the FTP control channel.
- rule 20 allows the FTP data channel for Passive FTP connections.

```
set firewall conntrack-tcp-loose disable

set firewall name eth1local rule 10 action accept
set firewall name eth1local rule 10 source address 192.168.40.56
set firewall name eth1local rule 10 destination address 192.168.40.1
set firewall name eth1local rule 10 protocol tcp
set firewall name eth1local rule 10 state established enable
set firewall name eth1local rule 10 state related enable
set firewall name eth1local rule 10 source port 21

set firewall name eth1local rule 20 action accept
set firewall name eth1local rule 20 source address 192.168.40.56
set firewall name eth1local rule 20 destination address 192.168.40.1
set firewall name eth1local rule 20 protocol tcp
set firewall name eth1local rule 20 state established enable
set firewall name eth1local rule 20 state related enable
set firewall name eth1local rule 20 source port 44440-44450

set interfaces ethernet eth1 firewall local name eth1local
```

3.9 Allow SCP from Vyatta itself to an internal SCP server and from an internal SCP client to Vyatta itself

You may want to save Vyatta's configuration to an internal SCP server or from an internal SCP client.

You can use WinSCP as the SCP client(if you've enabled the SSH service on Vyatta), see **Figure10**.

If you've enabled the SSH service on Vyatta itself, the already discussed above firewall rule to allow SSH to Vyatta will allow you to use an SCP client, and also allows you to use Vyatta as a SCP client and save the config to an internal SCP server, see **Figure11**, only this time we apply the firewall instance on the internal interface eth1.

```
set firewall conntrack-tcp-loose disable

set firewall name eth1local rule 10 action drop
set firewall name eth1local rule 10 protocol tcp
set firewall name eth1local rule 10 destination port 22
set firewall name eth1local rule 10 state new enable
set firewall name eth1local rule 10 recent count 4
set firewall name eth1local rule 10 recent time 60

set firewall name eth1local rule 20 action accept
set firewall name eth1local rule 20 protocol tcp
set firewall name eth1local rule 20 destination port 22
set firewall name eth1local rule 20 destination address 192.168.40.1
set firewall name eth1local rule 20 state new enable
set firewall name eth1local rule 20 state established enable
set firewall name eth1local rule 20 state related enable

set interfaces ethernet eth1 firewall local name eth1local
```

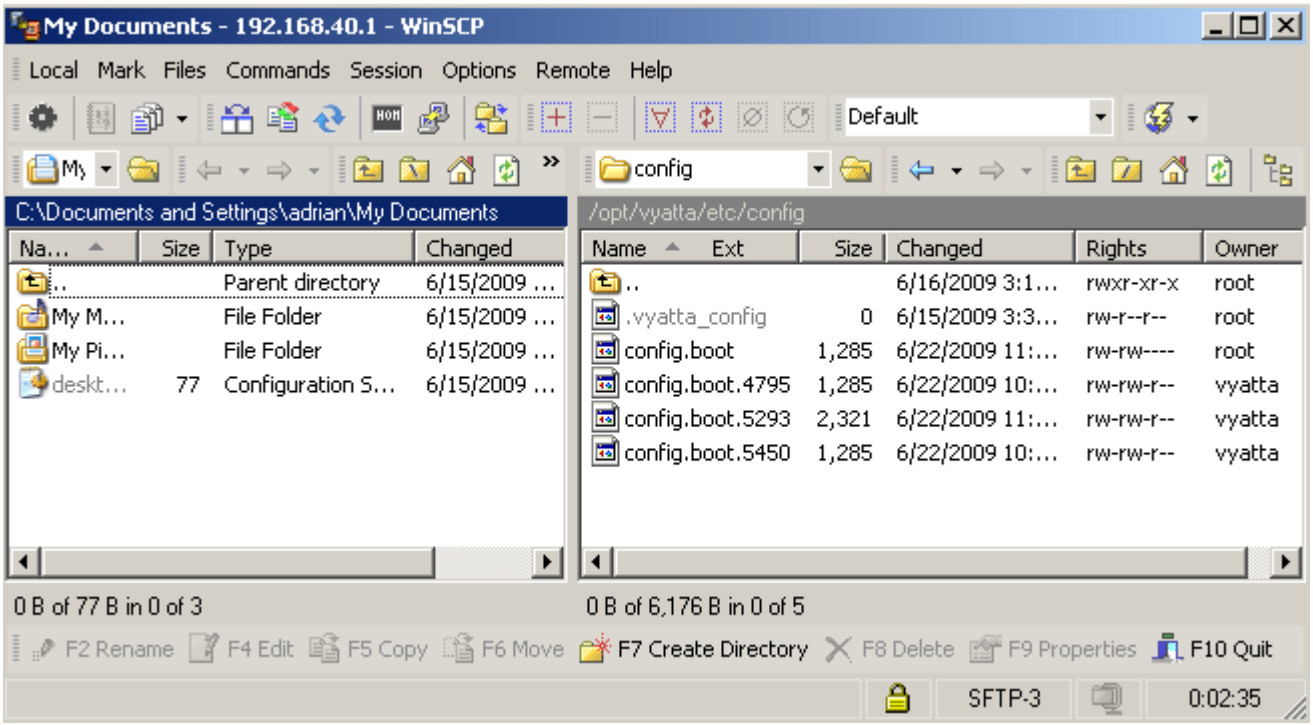

Figure10: WinSCP

```
vyatta@vyatta:/opt/vyatta/etc/config$ scp config.boot adrian@192.168.40.56:~
adrian@192.168.40.56's password:
config.boot                                   100% 1285     1.3KB/s   00:00
```

Figure11: Vyatta as SCP client

3.10 Allow Radius traffic from Vyatta itself to an internal Radius server

Say you're using an internal Radius server for VPN authentication, maybe an IAS server(domain integrated, on Windows 2003). Its IP address is 192.168.40.2.
- connections to UDP port 1812, Radius Authentication, rule 10.
- connections to UDP port 1813, Radius Accounting, rule 20.

```
set firewall name eth1local rule 10 action accept
set firewall name eth1local rule 10 protocol udp
set firewall name eth1local rule 10 source address 192.168.40.2
set firewall name eth1local rule 10 source port 1812
set firewall name eth1local rule 10 destination address 192.168.40.1
set firewall name eth1local rule 10 state established enable
set firewall name eth1local rule 10 state related enable

set firewall name eth1local rule 20 action accept
set firewall name eth1local rule 20 protocol udp
set firewall name eth1local rule 20 source address 192.168.40.2
set firewall name eth1local rule 20 source port 1813
set firewall name eth1local rule 20 destination address 192.168.40.1
set firewall name eth1local rule 20 state established enable
set firewall name eth1local rule 20 state related enable

set interfaces ethernet eth1 firewall local name eth1local
```

3.11 Vyatta as PPTP VPN Server: VPN traffic destined to Vyatta itself

Say Vyatta is acting as a PPTP VPN server.
So we need to allow on the Internet facing interface(the interface listening for incoming PPTP connections) connections to TCP port 1723 and GRE(IP protocol 47) -I haven't seen a need to allow new connections for GRE-, loaded modules on Vyatta, see **Figure12** and **Figure13**.

```
vyatta@vyatta:~$ zcat /proc/config.gz | grep -i pptp
CONFIG_NF_CONNTRACK_PPTP=m
CONFIG_NF_NAT_PPTP=m
vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$ cat /proc/modules | grep -i pptp
nf_nat_pptp 3840 0 - Live 0xd0c0f000
nf_conntrack_pptp 6916 1 nf_nat_pptp, Live 0xd0bf5000
nf_conntrack_proto_gre 5760 1 nf_conntrack_pptp, Live 0xd0bf2000
nf_nat_proto_gre 3204 1 nf_nat_pptp, Live 0xd0bcd000
nf_nat 18072 7 iptable_nat,nf_nat_pptp,nf_nat_h323,nf_nat_sip,nf_nat_proto_gre,n
f_nat_tftp,nf_nat_ftp, Live 0xd0be9000
nf_conntrack 61140 15 iptable_nat,xt_NOTRACK,nf_nat_pptp,nf_conntrack_pptp,nf_co
nntrack_proto_gre,nf_nat_h323,nf_conntrack_h323,nf_nat_sip,nf_conntrack_sip,nf_n
at_tftp,nf_nat_ftp,nf_nat,nf_conntrack_ipv4,nf_conntrack_tftp,nf_conntrack_ftp,
Live 0xd0bf9000
vyatta@vyatta:~$
```

Figure12: PPTP modules

```
vyatta@vyatta:~$ zcat /proc/config.gz | grep -i gre
CONFIG_NET_IPGRE=m
CONFIG_NET_IPGRE_BROADCAST=y
CONFIG_NF_CT_PROTO_GRE=m
CONFIG_NF_NAT_PROTO_GRE=m
CONFIG_NET_SCH_GRED=m
CONFIG_NET_SCH_INGRESS=m
vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$ cat /proc/modules | grep -i gre
nf_conntrack_proto_gre 5760 1 nf_conntrack_pptp, Live 0xd0bf2000
nf_nat_proto_gre 3204 1 nf_nat_pptp, Live 0xd0bcd000
nf_nat 18072 7 iptable_nat,nf_nat_pptp,nf_nat_h323,nf_nat_sip,nf_nat_proto_gre,n
f_nat_tftp,nf_nat_ftp, Live 0xd0be9000
nf_conntrack 61140 15 iptable_nat,xt_NOTRACK,nf_nat_pptp,nf_conntrack_pptp,nf_co
nntrack_proto_gre,nf_nat_h323,nf_conntrack_h323,nf_nat_sip,nf_conntrack_sip,nf_n
at_tftp,nf_nat_ftp,nf_nat,nf_conntrack_ipv4,nf_conntrack_tftp,nf_conntrack_ftp,
Live 0xd0bf9000
vyatta@vyatta:~$
```

Figure13: GRE modules

```
set firewall conntrack-tcp-loose disable

set firewall name eth0local rule 10 action accept
set firewall name eth0local rule 10 protocol tcp
set firewall name eth0local rule 10 destination port 1723
set firewall name eth0local rule 10 destination address 192.168.22.240
set firewall name eth0local rule 10 state new enable
set firewall name eth0local rule 10 state established enable
set firewall name eth0local rule 10 state related enable

set firewall name eth0local rule 20 action accept
```

```
set firewall name eth0local rule 20 protocol gre
set firewall name eth0local rule 20 destination address 192.168.22.240
set firewall name eth0local rule 20 state established enable
set firewall name eth0local rule 20 state related enable

set interfaces ethernet eth0 firewall local name eth0local
```

3.12 Vyatta as L2TP/IPsec VPN Server: VPN traffic destined to Vyatta itself

Say Vyatta is acting as a L2TP/IPsec VPN server.
So we need to allow on the Internet facing interface(the interface listening for incoming L2TP/IPsec connections:
- connections to UDP port 500.
- connections to UDP port 4500.
- ESP traffic.
- L2TP inside IPsec traffic.

```
set firewall name eth0local rule 10 action accept
set firewall name eth0local rule 10 protocol udp
set firewall name eth0local rule 10 destination port 500
set firewall name eth0local rule 10 destination address 192.168.22.240
set firewall name eth0local rule 10 state new enable
set firewall name eth0local rule 10 state established enable
set firewall name eth0local rule 10 state related enable

set firewall name eth0local rule 20 action accept
set firewall name eth0local rule 20 protocol udp
set firewall name eth0local rule 20 destination port 4500
set firewall name eth0local rule 20 destination address 192.168.22.240
set firewall name eth0local rule 20 state new enable
set firewall name eth0local rule 20 state established enable
set firewall name eth0local rule 20 state related enable

set firewall name eth0local rule 30 action accept
set firewall name eth0local rule 30 protocol esp
set firewall name eth0local rule 30 destination address 192.168.22.240
set firewall name eth0local rule 30 state new enable
set firewall name eth0local rule 30 state established enable
set firewall name eth0local rule 30 state related enable

set firewall name eth0local rule 40 action accept
set firewall name eth0local rule 40 protocol udp
set firewall name eth0local rule 40 destination port 1701
set firewall name eth0local rule 40 ipsec match-ipsec
set firewall name eth0local rule 40 destination address 192.168.22.240
set firewall name eth0local rule 40 state new enable
set firewall name eth0local rule 40 state established enable
set firewall name eth0local rule 40 state related enable

set interfaces ethernet eth0 firewall local name eth0local
```

3.13 Vyatta as an IPsec tunnel mode VPN gateway: s2s traffic destined to Vyatta itself

Say Vyatta is acting as a VPN gateway, terminating IPsec tunnel mode s2s connections, see **Figure14**.

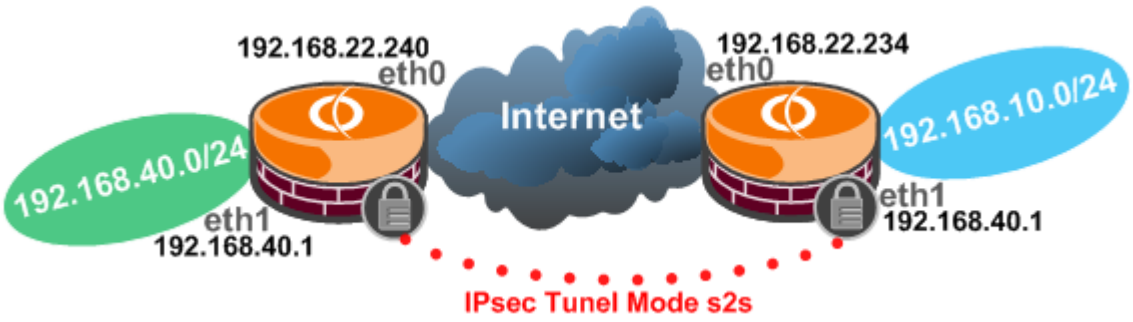

Figure14: Network Diagram

We need to allow s2s traffic destined to Vyatta itself, if the remote VPN gateway has a static IP address, you can specify it as source on the bellow rules:
- connections to UDP port 500, rule 10.
- ESP traffic, rule 20.
- possible connections to UDP port 4500, if there is a NAT device along the path between the two VPN gateways(rule 30 is needed only if there is a NAT device along the way, and if so then you don't need to allow ESP anymore).

```
set firewall name eth0local rule 10 action accept
set firewall name eth0local rule 10 protocol udp
set firewall name eth0local rule 10 destination port 500
set firewall name eth0local rule 10 destination address 192.168.22.240
set firewall name eth0local rule 10 state new enable
set firewall name eth0local rule 10 state established enable
set firewall name eth0local rule 10 state related enable

set firewall name eth0local rule 20 action accept
set firewall name eth0local rule 20 protocol esp
set firewall name eth0local rule 20 destination address 192.168.22.240
set firewall name eth0local rule 20 state new enable
set firewall name eth0local rule 20 state established enable
set firewall name eth0local rule 20 state related enable

set firewall name eth0local rule 30 action accept
set firewall name eth0local rule 30 protocol udp
set firewall name eth0local rule 30 destination port 4500
set firewall name eth0local rule 30 destination address 192.168.22.240
set firewall name eth0local rule 30 state new enable
set firewall name eth0local rule 30 state established enable
set firewall name eth0local rule 30 state related enable

set interfaces ethernet eth0 firewall local name eth0local
```

And on the Internet facing interface, eth0, we can add to the existing local firewall instance from above:

- traffic between the remote subnet and the IP address from the eth1 interface(traffic which is inside IPsec), because due to the existent local firewall instance on eth0 traffic coming from the remote site is denied, rule 30 for example allows SSH from remote management.
- if you want to ping from the local Vyatta to hosts on the remote site using ping with the -I parameter specifying the IP address of the eth1 interface(say /bin/ping -I 192.168.40.1 192.168.10.2), you need to permit ICMP echo-reply messages for established connections-inside IPsec-(connections for which the router has sent the ICMP echo-request message and is waiting the echo-reply message), rule 40.

- rule 41, is a little bit of a maniac/paranoid rule, just drops the echo-reply message from the remote subnets, messages which which was not received inside IPsec.
I've added it, although not quite needed, as, for example, it is possible to have a higher firewall number(>10) which allows returning echo-reply messages from the Internet to test connectivity from Vyatta itself, thus you may not specify a source address on this rule and so with the drop rule in place, we can avoid any overlapping, as "non-wanted" traffic-if any- will be dropped before it will reach that rule.
- if you want to ping the IP address the internal eth1 interface from the remote site, you need to allow the ICMP echo-request messages(inside IPsec), rule 50.

```
set firewall name eth0local rule 30 action accept
set firewall name eth0local rule 30 protocol tcp
set firewall name eth0local rule 30 destination port 22
set firewall name eth0local rule 30 destination address 192.168.40.1
set firewall name eth0local rule 30 source address 192.168.10.0/24
set firewall name eth0local rule 30 state new enable
set firewall name eth0local rule 30 state established enable
set firewall name eth0local rule 30 state related enable
set firewall name eth0local rule 30 ipsec match-ipsec

set firewall name eth0local rule 40 action accept
set firewall name eth0local rule 40 protocol icmp
set firewall name eth0local rule 40 source address 192.168.10.0/24
set firewall name eth0local rule 40 destination address 192.168.40.1
set firewall name eth0local rule 40 icmp type 0
set firewall name eth0local rule 40 icmp code 0
set firewall name eth0local rule 40 state established enable
set firewall name eth0local rule 40 state related enable
set firewall name eth0local rule 40 ipsec match-ipsec

set firewall name eth0local rule 41 action drop
set firewall name eth0local rule 41 protocol icmp
set firewall name eth0local rule 41 source address 192.168.10.0/24
set firewall name eth0local rule 41 destination address 192.168.40.1
set firewall name eth0local rule 41 icmp type 0
set firewall name eth0local rule 41 icmp code 0
set firewall name eth0local rule 41 state established enable
set firewall name eth0local rule 41 state related enable
set firewall name eth0local rule 41 ipsec match-ipsec

set firewall name eth0local rule 50 action accept
set firewall name eth0local rule 50 protocol icmp
set firewall name eth0local rule 50 source address 192.168.10.0/24
set firewall name eth0local rule 50 destination address 192.168.40.1
set firewall name eth0local rule 50 icmp type 8
set firewall name eth0local rule 50 icmp code 0
set firewall name eth0local rule 50 state new enable
set firewall name eth0local rule 50 state related enable
set firewall name eth0local rule 50 ipsec match-ipsec
```

3.14 GRE over IPsec: Traffic destined to Vyatta itself

We have to firewall traffic on the Internet facing interface(eth0), on the tunnel interface(tun1) and on the interface connected to the internal network(eth1), see **Figure15**.

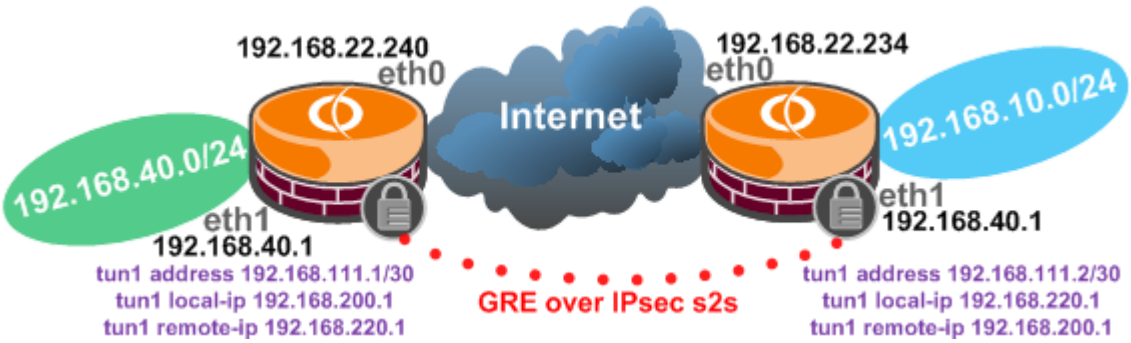

Figure15: Network Diagram

*** Local traffic on the Internet facing interface(eth0) ***
- traffic destined to UDP port 500, rule 10.
- ESP traffic, rule 20.
- GRE traffic inside IPsec, rule 30.

```
set firewall name eth0local rule 10 action accept
set firewall name eth0local rule 10 protocol udp
set firewall name eth0local rule 10 destination port 500
set firewall name eth0local rule 10 destination address 192.168.22.240
set firewall name eth0local rule 10 source address 192.168.22.234
set firewall name eth0local rule 10 state new enable
set firewall name eth0local rule 10 state established enable
set firewall name eth0local rule 10 state related enable

set firewall name eth0local rule 15 action accept
set firewall name eth0local rule 15 protocol esp
set firewall name eth0local rule 15 destination address 192.168.22.240
set firewall name eth0local rule 15 source address 192.168.22.234
set firewall name eth0local rule 15 state new enable
set firewall name eth0local rule 15 state established enable
set firewall name eth0local rule 15 state related enable

set firewall name eth0local rule 20 action accept
set firewall name eth0local rule 20 protocol gre
set firewall name eth0local rule 20 source address 192.168.220.1
set firewall name eth0local rule 20 destination address 192.168.200.1
set firewall name eth0local rule 20 ipsec match-ipsec
set firewall name eth0local rule 20 state new enable
set firewall name eth0local rule 20 state established enable
set firewall name eth0local rule 20 state related enable

set interfaces ethernet eth0 firewall local name eth0local
```

*** Local traffic on the tunnel interface(tun1) ***
- OSPF traffic, rule 10.
- if you want to manage the local Vyatta from the remote site using SSH(connection to the IP address of the eth1 interface), rule 20.
- if you want to ping from the local Vyatta to hosts on the remote site using ping with the -I parameter specifying the IP address of the eth1 interface(say /bin/ping -I 192.168.40.1 192.168.10.2), you need to permit ICMP echo-reply messages for established connections (connections for which the router has sent the ICMP echo-request message and is waiting the echo-reply message), rule 30.

- if you want to ping the IP address the internal eth1 interface from the remote site, you need to allow the ICMP echo-request messages, rule 40.

```
set firewall name tun1local rule 10 action accept
set firewall name tun1local rule 10 protocol 89
set firewall name tun1local rule 10 state new enable
set firewall name tun1local rule 10 state established enable
set firewall name tun1local rule 10 state related enable

set firewall name tun1local rule 20 action accept
set firewall name tun1local rule 20 protocol tcp
set firewall name tun1local rule 20 destination port 22
set firewall name tun1local rule 20 destination address 192.168.40.1
set firewall name tun1local rule 20 source address 192.168.10.0/24
set firewall name tun1local rule 20 state new enable
set firewall name tun1local rule 20 state established enable
set firewall name tun1local rule 20 state related enable

set firewall name tun1local rule 30 action accept
set firewall name tun1local rule 30 protocol icmp
set firewall name tun1local rule 30 source address 192.168.10.0/24
set firewall name tun1local rule 30 destination address 192.168.40.1
set firewall name tun1local rule 30 icmp type 0
set firewall name tun1local rule 30 icmp code 0
set firewall name tun1local rule 30 state established enable
set firewall name tun1local rule 30 state related enable

set firewall name tun1local rule 40 action accept
set firewall name tun1local rule 40 protocol icmp
set firewall name tun1local rule 40 source address 192.168.10.0/24
set firewall name tun1local rule 40 destination address 192.168.40.1
set firewall name tun1local rule 40 icmp type 8
set firewall name tun1local rule 40 icmp code 0
set firewall name tun1local rule 40 state new enable
set firewall name tun1local rule 40 state related enable

set interfaces tunnel tun1 firewall local name tun1local
```

*** Local traffic on the internal interface(eth1) ***
- OSPF traffic, rule 10.

```
set firewall name eth1local rule 10 action accept
set firewall name eth1local rule 10 protocol 89
set firewall name eth1local rule 10 state new enable
set firewall name eth1local rule 10 state established enable
set firewall name eth1local rule 10 state related enable

set interfaces ethernet eth1 firewall local name eth1local
```

3.15 Allow OSPF traffic to Vyatta itself

A basic firewall rule allowing OSPF(IP Protocol 89).

```
set firewall name eth1local rule 10 action accept
```

```
set firewall name eth1local rule 10 protocol 89
set firewall name eth1local rule 10 state new enable
set firewall name eth1local rule 10 state established enable
set firewall name eth1local rule 10 state related enable

set interfaces ethernet eth1 firewall local name eth1local
```

3.16 Allow RIP traffic to Vyatta itself

A basic firewall rule allowing RIP(RIP uses UDP port 520).

```
set firewall name eth1local rule 10 action accept
set firewall name eth1local rule 10 protocol udp
set firewall name eth1local rule 10 source port 520
set firewall name eth1local rule 10 destination port 520
set firewall name eth1local rule 10 state new enable
set firewall name eth1local rule 10 state established enable
set firewall name eth1local rule 10 state related enable

set interfaces ethernet eth1 firewall local name eth1local
```

3.17 Vyatta as DHCP server

Say you enable the DHCP server on Vyatta to server internal clients on the eth1 interface.

```
set firewall name eth1local rule 10 action accept
set firewall name eth1local rule 10 protocol udp
set firewall name eth1local rule 10 destination port 67
set firewall name eth1local rule 10 source port 68
set firewall name eth1local rule 10 state new enable
set firewall name eth1local rule 10 state established enable
set firewall name eth1local rule 10 state related enable

set interfaces ethernet eth1 firewall local name eth1local
```

3.18 Vyatta as DHCP relay server

Say you enable the DHCP relay on Vyatta to relay DHCP requests received from DHCP clients on the eth1 interface to a DHCP server located on the eth0 interface.

```
set firewall name eth1local rule 10 action accept
set firewall name eth1local rule 10 protocol udp
set firewall name eth1local rule 10 destination port 67
set firewall name eth1local rule 10 source port 68
set firewall name eth1local rule 10 state new enable
set firewall name eth1local rule 10 state established enable
set firewall name eth1local rule 10 state related enable

set interfaces ethernet eth1 firewall local name eth1local
```

Now, although the bellow firewall rule may appear logical, it does not have any effect(in fact I could have applied it as a general drop rule(delete the protocol, source etc.) and it would not drop the replies from the DHCP server):

```
set firewall name eth0local rule 10 action accept
```

```
set firewall name eth0local rule 10 protocol udp
set firewall name eth0local rule 10 destination port 67
set firewall name eth0local rule 10 source port 67
set firewall name eth0local rule 10 state established enable
set firewall name eth0local rule 10 state related enable

set interfaces ethernet eth0 firewall local name eth0local
```

3.19 DNS Forwarding on Vyatta

Say you configured Vyatta to act as a DNS relay for internal clients.
So internal clients have set as a DNS server Vyatta's IP address from its internal interface(eth1) and send their DNS queries to Vyatta, and Vyatta forwards these queries for example to the ISPs DNS servers.
Bellow we have assumed the ISP's DNS server is 192.168.22.1 and that Vyatta has a fixed IP address on the Internet facing interface.

```
set firewall name eth1local rule 10 action accept
set firewall name eth1local rule 10 protocol udp
set firewall name eth1local rule 10 source address 192.168.40.0/24
set firewall name eth1local rule 10 destination port 53
set firewall name eth1local rule 10 destination address 192.168.40.1
set firewall name eth1local rule 10 state new enable
set firewall name eth1local rule 10 state established enable
set firewall name eth1local rule 10 state related enable

set interfaces ethernet eth1 firewall local name eth1local

set firewall name eth0local rule 10 action accept
set firewall name eth0local rule 10 protocol udp
set firewall name eth0local rule 10 source address 192.168.22.1
set firewall name eth0local rule 10 source port 53
set firewall name eth0local rule 10 destination address 192.168.22.240
set firewall name eth0local rule 10 state established enable
set firewall name eth0local rule 10 state related enable

set interfaces ethernet eth0 firewall local name eth0local
```

3.20 Vyatta as a web proxy

Say you enable the web proxy on Vyatta to serve internal clients(you enable it on the internal interface eth1 listening on IP address 192.168.40.1 and on TCP port 8080).

Vyatta must be able to resolve DNS names for web proxy clients, so you need to configure on it a name server(or maybe you obtain from your ISP through DHCP your IP configuration and so it will query the DNS server provided like so).
When Vyatta's web proxy is in intercepting mode, clients not configured as web proxy clients must be able to resolve DNS names(so depeding on what DNS server(s) they query-Vyatta itself when it does DNS forwarding, ISP's DNS servers, internal DNS server etc.-, you will have to configure appropriatelly firewall rules).

First we allow the web proxy client's requests on the eth1 interface. Note that this rule will also "apply" for the requests of the clients not configured as web proxy clients when Vyatta's web proxy is in intercepting mode, sort of, meaning that you don't need to also allow as an **in** firewall instance on the eth1 interface the HTTP/HTTPS traffic of these clients:

```
set firewall conntrack-tcp-loose disable

set firewall name eth1local rule 10 action accept
set firewall name eth1local rule 10 protocol tcp
set firewall name eth1local rule 10 destination port 8080
set firewall name eth1local rule 10 destination address 192.168.40.1
set firewall name eth1local rule 10 state new enable
set firewall name eth1local rule 10 state established enable
set firewall name eth1local rule 10 state related enable

set interfaces ethernet eth1 firewall local name eth1local
```

Then we allow on the eth0 interface returning web traffic originated from Vyatta's web proxy and also DNS query replies for DNS queries made by Vyatta itself to the name server statically configured on it(in this case 192.168.22.1).

```
set firewall name eth0local rule 10 action accept
set firewall name eth0local rule 10 protocol tcp
set firewall name eth0local rule 10 source port 80,443
set firewall name eth0local rule 10 destination address 192.168.22.240
set firewall name eth0local rule 10 state established enable
set firewall name eth0local rule 10 state related enable

set firewall name eth0local rule 20 action accept
set firewall name eth0local rule 20 protocol udp
set firewall name eth0local rule 20 source address 192.168.22.1
set firewall name eth0local rule 20 source port 53
set firewall name eth0local rule 20 destination address 192.168.22.240
set firewall name eth0local rule 20 state established enable
set firewall name eth0local rule 20 state related enable

set interfaces ethernet eth0 firewall local name eth0local
```

4. Traffic through Vyatta

You may want to allow clients behind Vyatta to access external servers(DNS, HTTP/HTTPS, FTP etc.)

4.1 Allow FTP through Vyatta

Say you want to allow internal clients to connect through Vyatta to external FTP servers, see **Figure16**.

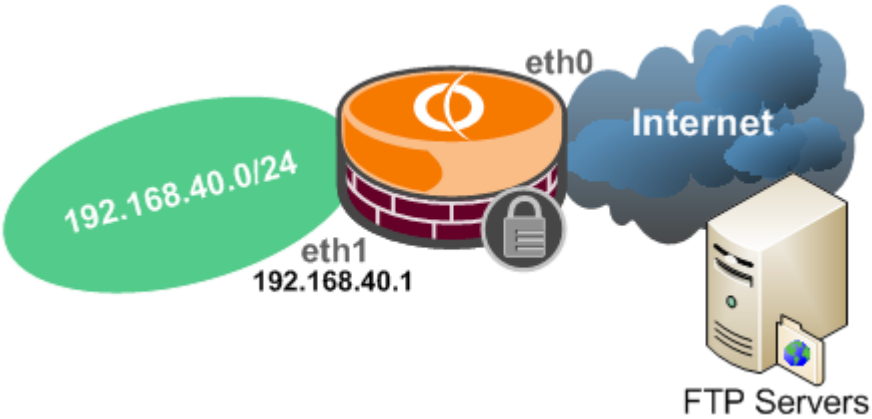

Figure16: Network Diagram

As we have previously noted we have on Vyatta the needed FTP modules, see **Figure17**.

```
vyatta@vyatta:~$ zcat /proc/config.gz | grep -i ftp
CONFIG_IP_VS_FTP=m
CONFIG_NF_CONNTRACK_FTP=m
CONFIG_NF_CONNTRACK_TFTP=m
CONFIG_NF_NAT_FTP=m
CONFIG_NF_NAT_TFTP=m
vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$ cat /proc/modules | grep -i ftp
nf_nat_tftp 2432 0 - Live 0xd0bca000
nf_nat_ftp 3712 0 - Live 0xd0bc8000
nf_nat 18072 7 iptable_nat,nf_nat_pptp,nf_nat_h323,nf_nat_sip,nf_nat_proto_gre,n
f_nat_tftp,nf_nat_ftp, Live 0xd0be8000
nf_conntrack_tftp 5140 1 nf_nat_tftp, Live 0xd0b8b000
nf_conntrack_ftp 8356 1 nf_nat_ftp, Live 0xd0bc4000
nf_conntrack 61140 15 iptable_nat,xt_NOTRACK,nf_nat_pptp,nf_conntrack_pptp,nf_co
nntrack_proto_gre,nf_nat_h323,nf_conntrack_h323,nf_nat_sip,nf_conntrack_sip,nf_n
at_tftp,nf_nat_ftp,nf_nat,nf_conntrack_ipv4,nf_conntrack_tftp,nf_conntrack_ftp,
Live 0xd0bf8000
vyatta@vyatta:~$
```

Figure17: Loaded FTP Modules

So you should be able to allow with no major problems Active and Passive FTP through Vyatta.

*** **In** firewall instance on the internal interface(eth1) ***
We need to allow FTP control and data channels.

```
set firewall conntrack-tcp-loose disable
```

Allow FTP control channel:

```
set firewall name eth1in rule 10 action accept
set firewall name eth1in rule 10 protocol tcp
set firewall name eth1in rule 10 source address 192.168.40.0/24
set firewall name eth1in rule 10 destination port 21
set firewall name eth1in rule 10 state new enable
set firewall name eth1in rule 10 state established enable
set firewall name eth1in rule 10 state related enable
```

Allow FTP data channel(ftp connection tracking, for passive FTP for example the module "opens" the needed port for you, also the PORT command issued by the client behind NAT for active FTP is "translated" accordingly, I've used the destination port address range on rule 20 because the common ports aren't supposed to be used on the server in this case) -here the multiple source and destination ports would have been useful-:
- rule 15 allows Active FTP
- rule 20 allows Passive FTP

```
set firewall name eth1in rule 15 action accept
set firewall name eth1in rule 15 protocol tcp
set firewall name eth1in rule 15 source address 192.168.40.0/24
set firewall name eth1in rule 15 destination port 20
set firewall name eth1in rule 15 state established enable
set firewall name eth1in rule 15 state related enable
```

```
set firewall name eth1in rule 20 action accept
set firewall name eth1in rule 20 protocol tcp
set firewall name eth1in rule 20 source address 192.168.40.0/24
set firewall name eth1in rule 20 destination port 1024-65535
set firewall name eth1in rule 20 state established enable
set firewall name eth1in rule 20 state related enable

set interfaces ethernet eth1 firewall in name eth1in
```

*** **Out** firewall instance on the internal interface(eth1) ***
We don't need to permit new connections for this firewall instance.

Allow FTP control channel:

```
set firewall name eth1out rule 10 action accept
set firewall name eth1out rule 10 protocol tcp
set firewall name eth1out rule 10 destination address 192.168.40.0/24
set firewall name eth1out rule 10 source port 21
set firewall name eth1out rule 10 state established enable
set firewall name eth1out rule 10 state related enable
```

Allow FTP data channel:
- rule 15 allows Active FTP
- rule 20 allows Passive FTP

```
set firewall name eth1out rule 15 action accept
set firewall name eth1out rule 15 protocol tcp
set firewall name eth1out rule 15 destination address 192.168.40.0/24
set firewall name eth1out rule 15 source port 20
set firewall name eth1out rule 15 state established enable
set firewall name eth1out rule 15 state related enable

set firewall name eth1out rule 20 action accept
set firewall name eth1out rule 20 protocol tcp
set firewall name eth1out rule 20 destination address 192.168.40.0/24
set firewall name eth1out rule 20 source port 1024-65535
set firewall name eth1out rule 20 state established enable
set firewall name eth1out rule 20 state related enable

set interfaces ethernet eth1 firewall out name eth1out
```

*** **In** firewall instance on the Internet facing interface(eth0) ***
We need to allow FTP control and data channels.
No need to allow new connections.

Allow FTP control channel:

```
set firewall name eth0in rule 10 action accept
set firewall name eth0in rule 10 protocol tcp
set firewall name eth0in rule 10 destination address 192.168.40.0/24
set firewall name eth0in rule 10 source port 21
set firewall name eth0in rule 10 state established enable
set firewall name eth0in rule 10 state related enable
```

Allow FTP data channel:
- rule 15 allows Active FTP
- rule 20 allows Passive FTP

```
set firewall name eth0in rule 15 action accept
set firewall name eth0in rule 15 protocol tcp
set firewall name eth0in rule 15 destination address 192.168.40.0/24
set firewall name eth0in rule 15 source port 20
set firewall name eth0in rule 15 state established enable
set firewall name eth0in rule 15 state related enable

set firewall name eth0in rule 20 action accept
set firewall name eth0in rule 20 protocol tcp
set firewall name eth0in rule 20 destination address 192.168.40.0/24
set firewall name eth0in rule 20 source port 1024-65535
set firewall name eth0in rule 20 state established enable
set firewall name eth0in rule 20 state related enable

set interfaces ethernet eth0 firewall in name eth0in
```

*** **Out** firewall instance on the Internet facing interface(eth0) ***
We need to permit new connections for this firewall instance only for the FTP control channel.

Allow FTP control channel:

```
set firewall name eth0out rule 10 action accept
set firewall name eth0out rule 10 protocol tcp
set firewall name eth0out rule 10 source address 192.168.40.0/24
set firewall name eth0out rule 10 destination port 21
set firewall name eth0out rule 10 state new enable
set firewall name eth0out rule 10 state established enable
set firewall name eth0out rule 10 state related enable
```

Allow FTP data channel:
- rule 15 allows Active FTP
- rule 20 allows Passive FTP

```
set firewall name eth0out rule 15 action accept
set firewall name eth0out rule 15 protocol tcp
set firewall name eth0out rule 15 source address 192.168.40.0/24
set firewall name eth0out rule 15 destination port 20
set firewall name eth0out rule 15 state established enable
set firewall name eth0out rule 15 state related enable

set firewall name eth0out rule 20 action accept
set firewall name eth0out rule 20 protocol tcp
set firewall name eth0out rule 20 source address 192.168.40.0/24
set firewall name eth0out rule 20 destination port 1024-65535
set firewall name eth0out rule 20 state established enable
set firewall name eth0out rule 20 state related enable

set interfaces ethernet eth0 firewall out name eth0out
```

If you may want to prevent a host behind Vyatta to SYN flood an external FTP server, you may add to the **in** firewall instance on eth1 (it drops new connections from a host if more that 20 new connections to the FTP control channel port were seen in 10 seconds, depending on your design, these limits may vary) -rule 5 is not quite the most brilliant from of protection tough -:) -:

```
set firewall name eth1in rule 5 action drop
set firewall name eth1in rule 5 protocol tcp
set firewall name eth1in rule 5 state new enable
set firewall name eth1in rule 5 destination port 21
set firewall name eth1in rule 5 recent count 20
set firewall name eth1in rule 5 recent time 10
```

4.2 Allow TFTP through Vyatta

Say you want to allow hosts behind a Vyatta interface(eth0) to access a TFTP server located behind another Vyatta interface(eth1), see **Figure18**.

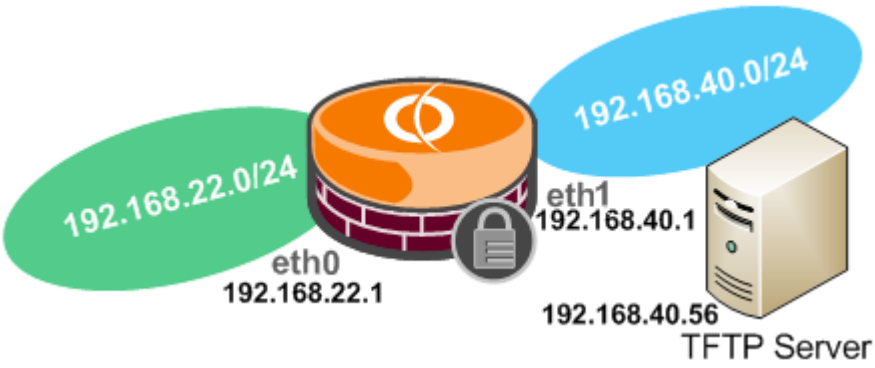

Figure18: Network Diagram

I will use TFtpd32 as the TFTP server.
For convenience I've specified a pool of local ports on the TFTP server, see **Figure19**.

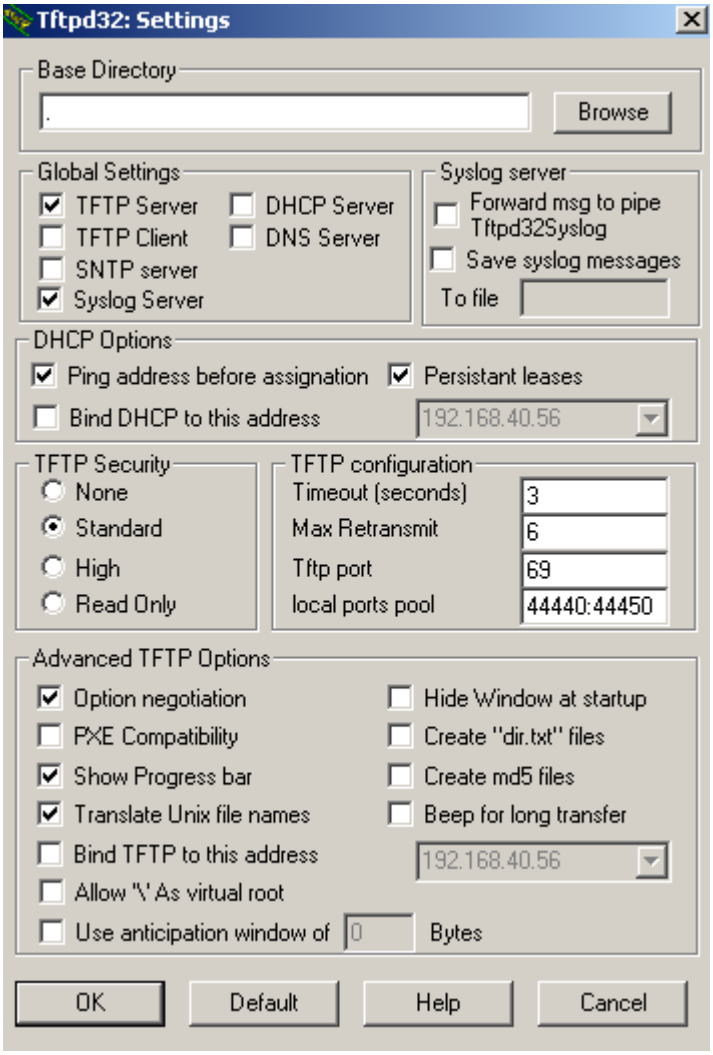

Figure19: TFTP Server settings

We have the needed TFTP modules loaded, see **Figure20**.

```
vyatta@vyatta:~$ zcat /proc/config.gz | grep -i tftp
CONFIG_NF_CONNTRACK_TFTP=m
CONFIG_NF_NAT_TFTP=m
vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$ cat /proc/modules | grep -i tftp
nf_nat_tftp 2432 0 - Live 0xd0bcb000
nf_nat 18072 7 iptable_nat,nf_nat_pptp,nf_nat_h323,nf_nat_sip,nf_nat_proto_gre,n
f_nat_tftp,nf_nat_ftp, Live 0xd0be9000
nf_conntrack_tftp 5140 1 nf_nat_tftp, Live 0xd0b76000
nf_conntrack 61140 15 iptable_nat,xt_NOTRACK,nf_nat_pptp,nf_conntrack_pptp,nf_co
nntrack_proto_gre,nf_nat_h323,nf_conntrack_h323,nf_nat_sip,nf_conntrack_sip,nf_n
at_tftp,nf_nat_ftp,nf_nat,nf_conntrack_ipv4,nf_conntrack_tftp,nf_conntrack_ftp,
Live 0xd0bf9000
vyatta@vyatta:~$
```

Figure20: Loaded TFTP modules

*** **In** firewall instance on the eth1 interface ***

```
set firewall name eth1in rule 10 action accept
set firewall name eth1in rule 10 protocol udp
```

```
set firewall name eth1in rule 10 source address 192.168.40.56
set firewall name eth1in rule 10 destination address 192.168.22.0/24
set firewall name eth1in rule 10 source port 44440-44450
set firewall name eth1in rule 10 state established enable
set firewall name eth1in rule 10 state related enable

set interfaces ethernet eth1 firewall in name eth1in
```

*** **Out** firewall instance on the eth1 interface ***

```
set firewall name eth1out rule 10 action accept
set firewall name eth1out rule 10 protocol udp
set firewall name eth1out rule 10 source address 192.168.22.0/24
set firewall name eth1out rule 10 destination address 192.168.40.56
set firewall name eth1out rule 10 destination port 69
set firewall name eth1out rule 10 state new enable
set firewall name eth1out rule 10 state established enable
set firewall name eth1out rule 10 state related enable

set firewall name eth1out rule 20 action accept
set firewall name eth1out rule 20 protocol udp
set firewall name eth1out rule 20 source address 192.168.22.0/24
set firewall name eth1out rule 20 destination address 192.168.40.56
set firewall name eth1out rule 20 destination port 44440-44450
set firewall name eth1out rule 20 state established enable
set firewall name eth1out rule 20 state related enable

set interfaces ethernet eth1 firewall out name eth1out
```

*** **In** firewall instance on the eth0 interface ***

```
set firewall name eth0in rule 10 action accept
set firewall name eth0in rule 10 protocol udp
set firewall name eth0in rule 10 source address 192.168.22.0/24
set firewall name eth0in rule 10 destination address 192.168.40.56
set firewall name eth0in rule 10 destination port 69
set firewall name eth0in rule 10 state new enable
set firewall name eth0in rule 10 state established enable
set firewall name eth0in rule 10 state related enable

set firewall name eth0in rule 20 action accept
set firewall name eth0in rule 20 protocol udp
set firewall name eth0in rule 20 source address 192.168.22.0/24
set firewall name eth0in rule 20 destination address 192.168.40.56
set firewall name eth0in rule 20 destination port 44440-44450
set firewall name eth0in rule 20 state established enable
set firewall name eth0in rule 20 state related enable

set interfaces ethernet eth0 firewall in name eth0in
```

*** **Out** firewall instance on the eth0 interface ***

```
set firewall name eth0out rule 10 action accept
```

```
set firewall name eth0out rule 10 protocol udp
set firewall name eth0out rule 10 source address 192.168.40.56
set firewall name eth0out rule 10 source port 44440-44450
set firewall name eth0out rule 10 destination address 192.168.22.0/24
set firewall name eth0out rule 10 state established enable
set firewall name eth0out rule 10 state related enable

set interfaces ethernet eth0 firewall out name eth0out
```

4.3 Allow web traffic through Vyatta

Say you may want to allow HTTP and HTTPS traffic through Vyatta when the web proxy is not enabled(if the web proxy is enabled search for Vyatta as web proxy in this article), see **Figure21**.

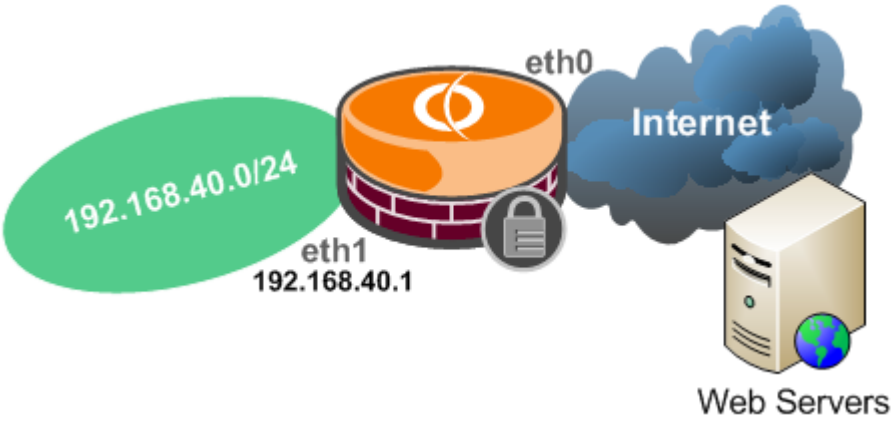

Figure21: Network Diagram

```
set firewall conntrack-tcp-loose disable
```

*** **In** firewall instance on the internal interface(eth1) ***
We need to permit new connections for this firewall instance.

```
set firewall name eth1in rule 10 action accept
set firewall name eth1in rule 10 protocol tcp
set firewall name eth1in rule 10 source address 192.168.40.0/24
set firewall name eth1in rule 10 destination port 80,443
set firewall name eth1in rule 10 state new enable
set firewall name eth1in rule 10 state established enable
set firewall name eth1in rule 10 state related enable

set interfaces ethernet eth1 firewall in name eth1in
```

*** **Out** firewall instance on the internal interface(eth1) ***
We don't need to permit new connections for this firewall instance.

```
set firewall name eth1out rule 10 action accept
set firewall name eth1out rule 10 protocol tcp
set firewall name eth1out rule 10 destination address 192.168.40.0/24
set firewall name eth1out rule 10 source port 80,443
set firewall name eth1out rule 10 state established enable
set firewall name eth1out rule 10 state related enable

set interfaces ethernet eth1 firewall out name eth1out
```

*** **In** firewall instance on the Internet facing interface(eth0) ***
We don't need to permit new connections for this firewall instance.

```
set firewall name eth0in rule 10 action accept
set firewall name eth0in rule 10 protocol tcp
set firewall name eth0in rule 10 destination address 192.168.40.0/24
set firewall name eth0in rule 10 source port 80,443
set firewall name eth0in rule 10 state established enable
set firewall name eth0in rule 10 state related enable

set interfaces ethernet eth0 firewall in name eth0in
```

*** **Out** firewall instance on the Internet facing interface(eth0) ***
We don't need to permit new connections for this firewall instance.

```
set firewall name eth0out rule 10 action accept
set firewall name eth0out rule 10 protocol tcp
set firewall name eth0out rule 10 source address 192.168.40.0/24
set firewall name eth0out rule 10 destination port 80,443
set firewall name eth0out rule 10 state new enable
set firewall name eth0out rule 10 state established enable
set firewall name eth0out rule 10 state related enable

set interfaces ethernet eth0 firewall out name eth0out
```

If you may want to prevent a host behind Vyatta to SYN flood an external HTTP/HTTPS, you may add to the **in** firewall instance on eth1 (it drops new connections from a host if more that 20 new connections to the destination TCP 80/443 ports were seen in 2 seconds, depending on your design, these limits may vary) -rule 5 is not the most brilliant form of protection though -:) -:

```
set firewall name eth1in rule 5 action drop
set firewall name eth1in rule 5 protocol tcp
set firewall name eth1in rule 5 state new enable
set firewall name eth1in rule 5 destination port 80
set firewall name eth1in rule 5 recent count 20
set firewall name eth1in rule 5 recent time 2

set firewall name eth1in rule 6 action drop
set firewall name eth1in rule 6 protocol tcp
set firewall name eth1in rule 6 state new enable
set firewall name eth1in rule 6 destination port 443
set firewall name eth1in rule 6 recent count 20
set firewall name eth1in rule 6 recent time 2
```

4.4 Allow DNS through Vyatta

Say you want to allow DNS through Vyatta from internal clients to an external DNS server(192.168.22.1), see **Figure22**.

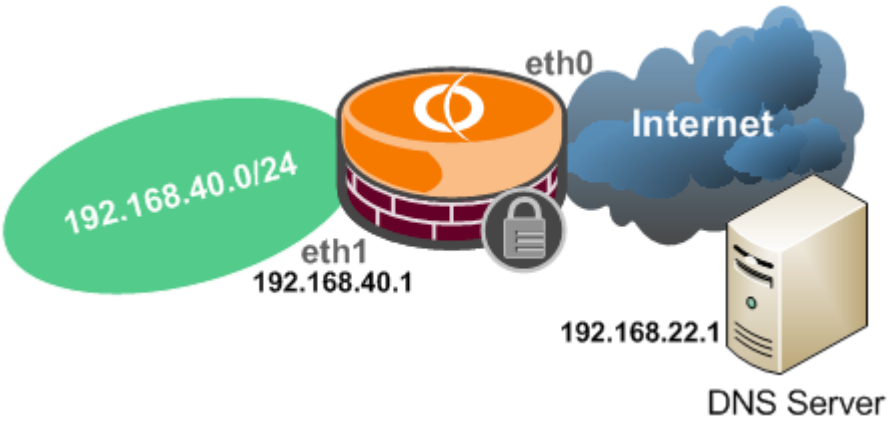

Figure22: Network Diagram

*** **In** firewall instance on the internal interface(eth1) ***

```
set firewall name eth1in rule 10 action accept
set firewall name eth1in rule 10 protocol udp
set firewall name eth1in rule 10 source address 192.168.40.0/24
set firewall name eth1in rule 10 destination port 53
set firewall name eth1in rule 10 destination address 192.168.22.1
set firewall name eth1in rule 10 state new enable
set firewall name eth1in rule 10 state established enable
set firewall name eth1in rule 10 state related enable

set interfaces ethernet eth1 firewall in name eth1in
```

*** **Out** firewall instance on the internal interface(eth1) ***

```
set firewall name eth1out rule 10 action accept
set firewall name eth1out rule 10 protocol udp
set firewall name eth1out rule 10 destination address 192.168.40.0/24
set firewall name eth1out rule 10 source port 53
set firewall name eth1out rule 10 source address 192.168.22.1
set firewall name eth1out rule 10 state established enable
set firewall name eth1out rule 10 state related enable

set interfaces ethernet eth1 firewall out name eth1out
```

*** **In** firewall instance on the Internet facing interface(eth0) ***

```
set firewall name eth0in rule 10 action accept
set firewall name eth0in rule 10 protocol udp
set firewall name eth0in rule 10 source address 192.168.22.1
set firewall name eth0in rule 10 source port 53
set firewall name eth0in rule 10 destination address 192.168.40.0/24
set firewall name eth0in rule 10 state established enable
set firewall name eth0in rule 10 state related enable

set interfaces ethernet eth0 firewall in name eth0in
```

*** **Out** firewall instance on the Internet facing interface(eth0) ***

```
set firewall name eth0out rule 10 action accept
set firewall name eth0out rule 10 protocol udp
set firewall name eth0out rule 10 source address 192.168.40.0/24
set firewall name eth0out rule 10 destination port 53
set firewall name eth0out rule 10 destination address 192.168.22.1
set firewall name eth0out rule 10 state new enable
set firewall name eth0out rule 10 state established enable
set firewall name eth0out rule 10 state related enable

set interfaces ethernet eth0 firewall out name eth0out
```

4.5 Allow Ping through Vyatta

Say you have internal client(s) behind Vyatta needing to ping external machines for some reasons or so(remember that ICMP can be "altered" to act as conduit for "bad" purposes, say as covert channel).

*** **In** firewall instance on the internal interface(eth1) ***
- allow ICMP echo request messages

```
set firewall name eth1in rule 10 action accept
set firewall name eth1in rule 10 protocol icmp
set firewall name eth1in rule 10 source address 192.168.40.0/24
set firewall name eth1in rule 10 icmp type 8
set firewall name eth1in rule 10 icmp code 0
set firewall name eth1in rule 10 state new enable
set firewall name eth1in rule 10 state established enable
set firewall name eth1in rule 10 state related enable

set interfaces ethernet eth1 firewall in name eth1in
```

*** **Out** firewall instance on the internal interface(eth1) ***
- allow ICMP echo reply messages

```
set firewall name eth1out rule 10 action accept
set firewall name eth1out rule 10 protocol icmp
set firewall name eth1out rule 10 destination address 192.168.40.0/24
set firewall name eth1out rule 10 icmp type 0
set firewall name eth1out rule 10 icmp code 0
set firewall name eth1out rule 10 state established enable
set firewall name eth1out rule 10 state related enable

set interfaces ethernet eth1 firewall out name eth1out
```

*** **In** firewall instance on the Internet facing interface(eth0) ***
- allow ICMP echo reply messages

```
set firewall name eth0in rule 10 action accept
set firewall name eth0in rule 10 protocol icmp
set firewall name eth0in rule 10 destination address 192.168.40.0/24
set firewall name eth0in rule 10 icmp type 0
set firewall name eth0in rule 10 icmp code 0
set firewall name eth0in rule 10 state established enable
set firewall name eth0in rule 10 state related enable
```

```
set interfaces ethernet eth0 firewall in name eth0in
```

*** **Out** firewall instance on the Internet facing interface(eth0) ***
- allow ICMP echo request messages

```
set firewall name eth0out rule 10 action accept
set firewall name eth0out rule 10 protocol icmp
set firewall name eth0out rule 10 source address 192.168.40.0/24
set firewall name eth0out rule 10 icmp type 8
set firewall name eth0out rule 10 icmp code 0
set firewall name eth0out rule 10 state new enable
set firewall name eth0out rule 10 state established enable
set firewall name eth0out rule 10 state related enable

set interfaces ethernet eth0 firewall out name eth0out
```

4.6 Allow PPTP through Vyatta

Say you have internal client(s) behind Vyatta needing to connect to an external PPTP server(192.168.22.234), see **Figure23**.

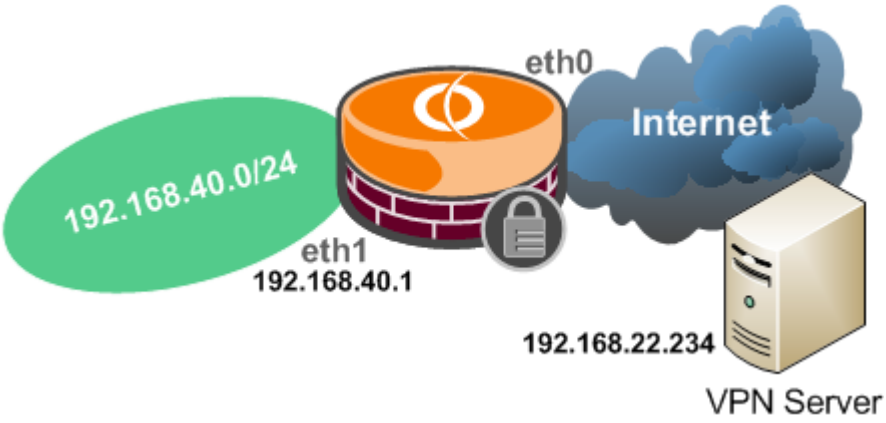

Figure23: Network Diagram

You can have multiple PPTP clients behind Vyatta connected to the same external PPTP server at a time. PPTP needs TCP ports 1723 and GRE(IP protocol 47).

```
set firewall conntrack-tcp-loose disable
```

As we have already discussed, the needed modules have been loaded, see **Figure24** and **Figure25**(and you don't need to allow new GRE connection through Vyatta).

```
vyatta@vyatta:~$ zcat /proc/config.gz | grep -i pptp
CONFIG_NF_CONNTRACK_PPTP=m
CONFIG_NF_NAT_PPTP=m
vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$ cat /proc/modules | grep -i pptp
nf_nat_pptp 3840 0 - Live 0xd0c0f000
nf_conntrack_pptp 6916 1 nf_nat_pptp, Live 0xd0bf5000
nf_conntrack_proto_gre 5760 1 nf_conntrack_pptp, Live 0xd0bf2000
nf_nat_proto_gre 3204 1 nf_nat_pptp, Live 0xd0bcd000
nf_nat 18072 7 iptable_nat,nf_nat_pptp,nf_nat_h323,nf_nat_sip,nf_nat_proto_gre,n
f_nat_tftp,nf_nat_ftp, Live 0xd0be9000
nf_conntrack 61140 15 iptable_nat,xt_NOTRACK,nf_nat_pptp,nf_conntrack_pptp,nf_co
nntrack_proto_gre,nf_nat_h323,nf_conntrack_h323,nf_nat_sip,nf_conntrack_sip,nf_n
at_tftp,nf_nat_ftp,nf_nat,nf_conntrack_ipv4,nf_conntrack_tftp,nf_conntrack_ftp,
Live 0xd0bf9000
vyatta@vyatta:~$
```

Figure24: PPTP modules

```
vyatta@vyatta:~$ zcat /proc/config.gz | grep -i gre
CONFIG_NET_IPGRE=m
CONFIG_NET_IPGRE_BROADCAST=y
CONFIG_NF_CT_PROTO_GRE=m
CONFIG_NF_NAT_PROTO_GRE=m
CONFIG_NET_SCH_GRED=m
CONFIG_NET_SCH_INGRESS=m
vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$ cat /proc/modules | grep -i gre
nf_conntrack_proto_gre 5760 1 nf_conntrack_pptp, Live 0xd0bf2000
nf_nat_proto_gre 3204 1 nf_nat_pptp, Live 0xd0bcd000
nf_nat 18072 7 iptable_nat,nf_nat_pptp,nf_nat_h323,nf_nat_sip,nf_nat_proto_gre,n
f_nat_tftp,nf_nat_ftp, Live 0xd0be9000
nf_conntrack 61140 15 iptable_nat,xt_NOTRACK,nf_nat_pptp,nf_conntrack_pptp,nf_co
nntrack_proto_gre,nf_nat_h323,nf_conntrack_h323,nf_nat_sip,nf_conntrack_sip,nf_n
at_tftp,nf_nat_ftp,nf_nat,nf_conntrack_ipv4,nf_conntrack_tftp,nf_conntrack_ftp,
Live 0xd0bf9000
vyatta@vyatta:~$
```

Figure25: GRE modules

*** **In** firewall instance on the internal interface(eth1) ***

set firewall name eth1in rule 10 action accept
set firewall name eth1in rule 10 protocol tcp
set firewall name eth1in rule 10 source address 192.168.40.0/24
set firewall name eth1in rule 10 destination port 1723
set firewall name eth1in rule 10 destination address 192.168.22.234
set firewall name eth1in rule 10 state new enable
set firewall name eth1in rule 10 state established enable
set firewall name eth1in rule 10 state related enable

set firewall name eth1in rule 20 action accept
set firewall name eth1in rule 20 protocol gre
set firewall name eth1in rule 20 source address 192.168.40.0/24
set firewall name eth1in rule 20 destination address 192.168.22.234
set firewall name eth1in rule 20 state established enable

set firewall name eth1in rule 20 state related enable

set interfaces ethernet eth1 firewall in name eth1in

*** **Out** firewall instance on the internal interface(eth1) ***

set firewall name eth1out rule 10 action accept
set firewall name eth1out rule 10 protocol tcp
set firewall name eth1out rule 10 destination address 192.168.40.0/24
set firewall name eth1out rule 10 source port 1723
set firewall name eth1out rule 10 source address 192.168.22.234
set firewall name eth1out rule 10 state established enable
set firewall name eth1out rule 10 state related enable

set firewall name eth1out rule 20 action accept
set firewall name eth1out rule 20 protocol gre
set firewall name eth1out rule 20 destination address 192.168.40.0/24
set firewall name eth1out rule 20 source address 192.168.22.234
set firewall name eth1out rule 20 state established enable
set firewall name eth1out rule 20 state related enable

set interfaces ethernet eth1 firewall out name eth1out

*** **In** firewall instance on the Internet facing interface(eth0) ***

set firewall name eth0in rule 10 action accept
set firewall name eth0in rule 10 protocol tcp
set firewall name eth0in rule 10 destination address 192.168.40.0/24
set firewall name eth0in rule 10 source port 1723
set firewall name eth0in rule 10 source address 192.168.22.234
set firewall name eth0in rule 10 state established enable
set firewall name eth0in rule 10 state related enable

set firewall name eth0in rule 20 action accept
set firewall name eth0in rule 20 protocol gre
set firewall name eth0in rule 20 destination address 192.168.40.0/24
set firewall name eth0in rule 20 source address 192.168.22.234
set firewall name eth0in rule 20 state established enable
set firewall name eth0in rule 20 state related enable

set interfaces ethernet eth0 firewall in name eth0in

*** **Out** firewall instance on the Internet facing interface(eth0) ***

set firewall name eth0out rule 10 action accept
set firewall name eth0out rule 10 protocol tcp
set firewall name eth0out rule 10 source address 192.168.40.0/24
set firewall name eth0out rule 10 destination port 1723
set firewall name eth0out rule 10 destination address 192.168.22.234
set firewall name eth0out rule 10 state new enable
set firewall name eth0out rule 10 state established enable
set firewall name eth0out rule 10 state related enable

```
set firewall name eth0out rule 20 action accept
set firewall name eth0out rule 20 protocol gre
set firewall name eth0out rule 20 source address 192.168.40.0/24
set firewall name eth0out rule 20 destination address 192.168.22.234
set firewall name eth0out rule 20 state established enable
set firewall name eth0out rule 20 state related enable

set interfaces ethernet eth0 firewall out name eth0out
```

4.7 Allow L2TP/IPsec through Vyatta

Say you have internal client(s) behind Vyatta needing to connect to an external L2TP/IPsec server(192.168.22.234), see **Figure26**.

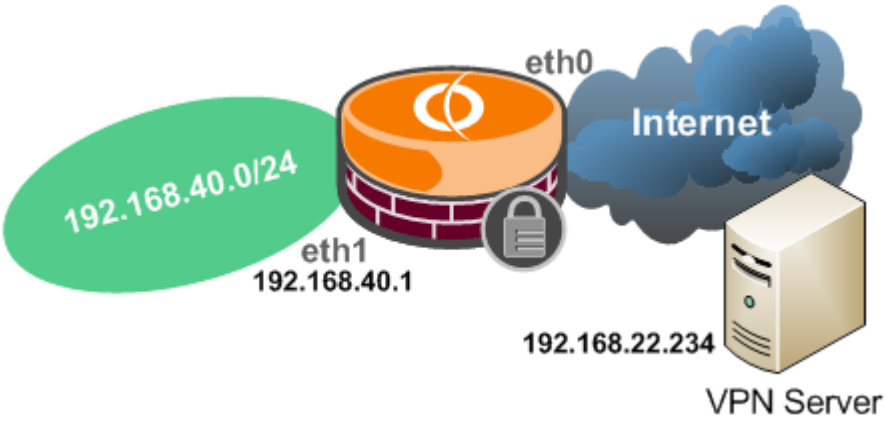

Figure26: Network Diagram

You can have multiple L2TP/IPsec clients behind Vyatta connected to the same external L2TP/IPsec server at a time.
L2TP/IPsec needs UDP ports 500 and 4500 if there is a NAT device between the client and the server along the path, otherwise it needs UDP port 500 and ESP(IP protocol 50).
To exemplify that, I've allowed both UDP port 4500 and ESP(rules 20 and 30), but you should allow what you need, probably UDP port 4500 as likely Vyatta itself will NAT.

*** **In** firewall instance on the internal interface(eth1) ***

```
set firewall name eth1in rule 10 action accept
set firewall name eth1in rule 10 protocol udp
set firewall name eth1in rule 10 source address 192.168.40.0/24
set firewall name eth1in rule 10 destination port 500
set firewall name eth1in rule 10 destination address 192.168.22.234
set firewall name eth1in rule 10 state new enable
set firewall name eth1in rule 10 state established enable
set firewall name eth1in rule 10 state related enable

set firewall name eth1in rule 20 action accept
set firewall name eth1in rule 20 protocol udp
set firewall name eth1in rule 20 destination port 4500
set firewall name eth1in rule 20 source address 192.168.40.0/24
set firewall name eth1in rule 20 destination address 192.168.22.234
set firewall name eth1in rule 20 state new enable
set firewall name eth1in rule 20 state established enable
set firewall name eth1in rule 20 state related enable
```

```
set firewall name eth1in rule 30 action accept
set firewall name eth1in rule 30 protocol esp
set firewall name eth1in rule 30 source address 192.168.40.0/24
set firewall name eth1in rule 30 destination address 192.168.22.234
set firewall name eth1in rule 30 state new enable
set firewall name eth1in rule 30 state established enable
set firewall name eth1in rule 30 state related enable

set interfaces ethernet eth1 firewall in name eth1in
```

*** **Out** firewall instance on the internal interface(eth1) ***

```
set firewall name eth1out rule 10 action accept
set firewall name eth1out rule 10 protocol udp
set firewall name eth1out rule 10 destination address 192.168.40.0/24
set firewall name eth1out rule 10 source port 500
set firewall name eth1out rule 10 source address 192.168.22.234
set firewall name eth1out rule 10 state established enable
set firewall name eth1out rule 10 state related enable

set firewall name eth1out rule 20 action accept
set firewall name eth1out rule 20 protocol udp
set firewall name eth1out rule 20 destination address 192.168.40.0/24
set firewall name eth1out rule 20 source port 4500
set firewall name eth1out rule 20 source address 192.168.22.234
set firewall name eth1out rule 20 state established enable
set firewall name eth1out rule 20 state related enable

set firewall name eth1out rule 30 action accept
set firewall name eth1out rule 30 protocol esp
set firewall name eth1out rule 30 destination address 192.168.40.0/24
set firewall name eth1out rule 30 source address 192.168.22.234
set firewall name eth1out rule 30 state established enable
set firewall name eth1out rule 30 state related enable

set interfaces ethernet eth1 firewall out name eth1out
```

*** **In** firewall instance on the Internet facing interface(eth0) ***

```
set firewall name eth0in rule 10 action accept
set firewall name eth0in rule 10 protocol udp
set firewall name eth0in rule 10 destination address 192.168.40.0/24
set firewall name eth0in rule 10 source port 500
set firewall name eth0in rule 10 source address 192.168.22.234
set firewall name eth0in rule 10 state established enable
set firewall name eth0in rule 10 state related enable

set firewall name eth0in rule 20 action accept
set firewall name eth0in rule 20 protocol udp
set firewall name eth0in rule 20 destination address 192.168.40.0/24
set firewall name eth0in rule 20 source port 4500
set firewall name eth0in rule 20 source address 192.168.22.234
set firewall name eth0in rule 20 state established enable
```

```
set firewall name eth0in rule 20 state related enable

set firewall name eth0in rule 30 action accept
set firewall name eth0in rule 30 protocol esp
set firewall name eth0in rule 30 destination address 192.168.40.0/24
set firewall name eth0in rule 30 source address 192.168.22.234
set firewall name eth0in rule 30 state established enable
set firewall name eth0in rule 30 state related enable

set interfaces ethernet eth0 firewall in name eth0in
```

*** **Out** firewall instance on the Internet facing interface(eth0) ***

```
set firewall name eth0out rule 10 action accept
set firewall name eth0out rule 10 protocol udp
set firewall name eth0out rule 10 source address 192.168.40.0/24
set firewall name eth0out rule 10 destination port 500
set firewall name eth0out rule 10 destination address 192.168.22.234
set firewall name eth0out rule 10 state new enable
set firewall name eth0out rule 10 state established enable
set firewall name eth0out rule 10 state related enable

set firewall name eth0out rule 20 action accept
set firewall name eth0out rule 20 protocol udp
set firewall name eth0out rule 20 source address 192.168.40.0/24
set firewall name eth0out rule 20 destination port 4500
set firewall name eth0out rule 20 state new enable
set firewall name eth0out rule 20 destination address 192.168.22.234
set firewall name eth0out rule 20 state established enable
set firewall name eth0out rule 20 state related enable

set firewall name eth0out rule 30 action accept
set firewall name eth0out rule 30 protocol esp
set firewall name eth0out rule 30 source address 192.168.40.0/24
set firewall name eth0out rule 30 destination address 192.168.22.234
set firewall name eth0out rule 30 state new enable
set firewall name eth0out rule 30 state established enable
set firewall name eth0out rule 30 state related enable

set interfaces ethernet eth0 firewall out name eth0out
```

4.8 Vyatta as L2TP/IPsec or PPTP VPN Server: Filter VPN clients' traffic

Say Vyatta is acting as a L2TP/IPsec or PPTP VPN server.
And you you may want to filter the VPN client's traffic, say what they can access on the internal network, or perform stateful packet inspection over their Internet traffic(in case split tunneling is not used).
Currently you cannot filter per VPN user/group traffic. Unless, maybe if you use a Radius server for authentication, and assign to each user a specific IP address.

Let's create some test rules.
The 192.168.40.220-192.168.40.230 range is the range of IP addresses from which VPN clients receive IP addresses for their PPP interface, see **Figure27**.

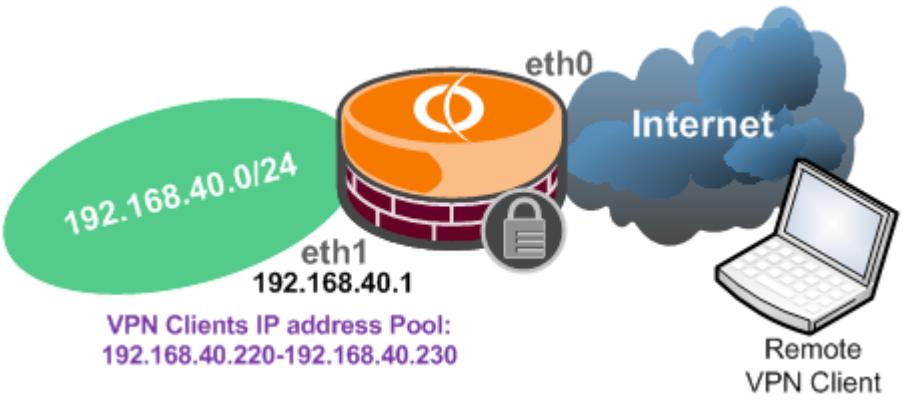

Figure27: Network Diagram

*** **In** firewall instance on the internal interface(eth1) ***
- allow HTTP/HTTPS to an internal server(192.168.40.10), rule 10.
- allow all traffic to an internal server(192.168.40.2), rule 20.

```
set firewall name eth1in rule 10 action accept
set firewall name eth1in rule 10 protocol tcp
set firewall name eth1in rule 10 source address 192.168.40.10
set firewall name eth1in rule 10 source port 80,443
set firewall name eth1in rule 10 destination address 192.168.40.220-192.168.40.230
set firewall name eth1in rule 10 state established enable
set firewall name eth1in rule 10 state related enable

set firewall name eth1in rule 20 action accept
set firewall name eth1in rule 20 protocol all
set firewall name eth1in rule 20 source address 192.168.40.2
set firewall name eth1in rule 20 destination address 192.168.40.220-192.168.40.230
set firewall name eth1in rule 20 state established enable
set firewall name eth1in rule 20 state related enable

set interfaces ethernet eth1 firewall in name eth1in
```

*** **Out** firewall instance on the internal interface(eth1) ***
- allow HTTP/HTTPS to an internal server(192.168.40.10), rule 10.
- allow all traffic to an internal server(192.168.40.2), rule 20.

```
set firewall name eth1out rule 10 action accept
set firewall name eth1out rule 10 protocol tcp
set firewall name eth1out rule 10 source address 192.168.40.220-192.168.40.230
set firewall name eth1out rule 10 destination port 80,443
set firewall name eth1out rule 10 destination address 192.168.40.10
set firewall name eth1out rule 10 state new enable
set firewall name eth1out rule 10 state established enable
set firewall name eth1out rule 10 state related enable

set firewall name eth1out rule 20 action accept
set firewall name eth1out rule 20 protocol all
set firewall name eth1out rule 20 source address 192.168.40.220-192.168.40.230
set firewall name eth1out rule 20 destination address 192.168.40.2
set firewall name eth1out rule 20 state new enable
set firewall name eth1out rule 20 state established enable
```

```
set firewall name eth1out rule 20 state related enable
set firewall name eth1out rule 20 state invalid disable

set interfaces ethernet eth1 firewall out name eth1out
```

*** **In** firewall instance on the Internet facing interface(eth0) ***
- stateful packet inspection over VPN clients' HTTP/HTTPS traffic destined to the Internet, rule 10.

```
set firewall name eth0in rule 10 action accept
set firewall name eth0in rule 10 protocol tcp
set firewall name eth0in rule 10 source port 80,443
set firewall name eth0in rule 10 destination address 192.168.40.220-192.168.40.230
set firewall name eth0in rule 10 state established enable
set firewall name eth0in rule 10 state related enable

set interfaces ethernet eth0 firewall in name eth0in
```

*** **Out** firewall instance on the Internet facing interface(eth0) ***
- stateful packet inspection over VPN clients' HTTP/HTTPS traffic destined to the Internet, rule 10.

```
set firewall name eth0out rule 10 action accept
set firewall name eth0out rule 10 protocol tcp
set firewall name eth0out rule 10 source address 192.168.40.220-192.168.40.230
set firewall name eth0out rule 10 destination port 80,443
set firewall name eth0out rule 10 state new enable
set firewall name eth0out rule 10 state established enable
set firewall name eth0out rule 10 state related enable

set interfaces ethernet eth0 firewall out name eth0out
```

4.9 Vyatta as an IPsec tunnel mode VPN gateway: s2s traffic between the local and remote subnets and vice-versa

We want to filter s2s traffic between the local and remote subnet and vice-versa, see **Figure28**.

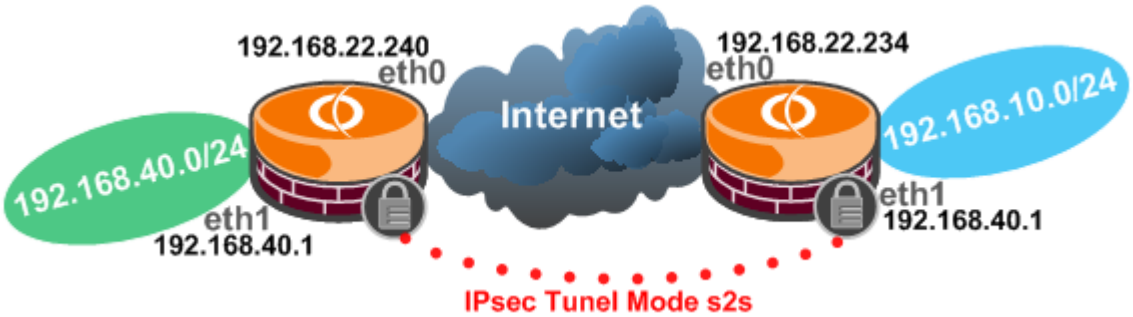

Figure28: Network Diagram

```
set firewall conntrack-tcp-loose disable
```

*** **In** firewall instance on the Internet facing interface(eth0) ***
- traffic between remote and local subnets, traffic which should be received inside IPsec, rule 10 basically allows all traffic.
- rule 11, is a little bit of a maniac/paranoid rule, just drops traffic between remote and local subnets, traffic which was not received inside IPsec.
I've added it, although not quite needed, as, for example, it is possible to have a higher firewall number(>10)

which allows returning traffic from the Internet for HTTP/HTTPS connections, thus you may not specify a source address on this rule and so with the drop rule in place, we can avoid any overlapping, as "non-wanted" traffic-if any- will be dropped before it will reach that rule.

```
set firewall name eth0in rule 10 action accept
set firewall name eth0in rule 10 source address 192.168.10.0/24
set firewall name eth0in rule 10 destination address 192.168.40.0/24
set firewall name eth0in rule 10 state new enable
set firewall name eth0in rule 10 state established enable
set firewall name eth0in rule 10 state related enable
set firewall name eth0in rule 10 ipsec match-ipsec

set firewall name eth0in rule 11 action drop
set firewall name eth0in rule 11 source address 192.168.10.0/24
set firewall name eth0in rule 11 destination address 192.168.40.0/24
set firewall name eth0in rule 11 state new enable
set firewall name eth0in rule 11 state established enable
set firewall name eth0in rule 11 state related enable

set interfaces ethernet eth0 firewall in name eth0in
```

*** **Out** firewall instance on the Internet facing interface(eth0) ***
- traffic between local and remote subnets(traffic which is not yet inside IPsec), rule 10.

```
set firewall name eth0out rule 10 action accept
set firewall name eth0out rule 10 source address 192.168.40.0/24
set firewall name eth0out rule 10 destination address 192.168.10.0/24
set firewall name eth0out rule 10 state new enable
set firewall name eth0out rule 10 state established enable
set firewall name eth0out rule 10 state related enable

set interfaces ethernet eth0 firewall out name eth0out
```

*** **In** firewall instance on the internal interface(eth1) ***
- traffic between local and remote subnets(traffic which is not yet inside IPsec), rule 10.

```
set firewall name eth1in rule 10 action accept
set firewall name eth1in rule 10 source address 192.168.40.0/24
set firewall name eth1in rule 10 destination address 192.168.10.0/24
set firewall name eth1in rule 10 state new enable
set firewall name eth1in rule 10 state established enable
set firewall name eth1in rule 10 state related enable

set interfaces ethernet eth1 firewall in name eth1in
```

*** **Out** firewall instance on the internal interface(eth1) ***
- traffic between remote and local subnets, traffic which should be received inside IPsec, rule 10 basically allows all traffic.
- rule 11, is a little bit of a maniac/paranoid rule, just drops traffic between remote and local subnets, traffic which was not received inside IPsec.
I've added it, although not quite needed, as, for example, it is possible to have a higher firewall number(>10) which allows returning traffic from the Internet for HTTP/HTTPS connections, thus you may not specify a source address on this rule and so with the drop rule in place, we can avoid any overlapping, as "non-wanted"

traffic-if any- will be dropped before it will reach that rule.

```
set firewall name eth1out rule 10 action accept
set firewall name eth1out rule 10 source address 192.168.10.0/24
set firewall name eth1out rule 10 destination address 192.168.40.0/24
set firewall name eth1out rule 10 state new enable
set firewall name eth1out rule 10 state established enable
set firewall name eth1out rule 10 state related enable
set firewall name eth1out rule 10 ipsec match-ipsec

set firewall name eth1out rule 11 action drop
set firewall name eth1out rule 11 source address 192.168.10.0/24
set firewall name eth1out rule 11 destination address 192.168.40.0/24
set firewall name eth1out rule 11 state new enable
set firewall name eth1out rule 11 state established enable
set firewall name eth1out rule 11 state related enable

set interfaces ethernet eth1 firewall out name eth1out
```

4.10 Vyatta as an IPsec tunnel mode VPN gateway: Excluding from the NAT process traffic destined to the remote subnet(s)

You may have on the eth0 interface a source or masquerade NAT rule. And this rule may apply to the traffic destined to the remote subnet(s). See **Figure28**.

There are a couple of ways to make the needed exclusion(s).
My favourite is to use the **exclude** parameter on the NAT rules.

For a masquerade NAT rule on eth0:
- rule 10 excludes from the NAT process traffic from the local subnet destined to the remote site(if you have multiple local or remote subnets, add as needed 11, 12, etc. NAT exclusions rules).
- rule 20 is the masquerade NAT type rule that NATs traffic.

```
set service nat rule 10 type masquerade
set service nat rule 10 source address 192.168.40.0/24
set service nat rule 10 destination address 192.168.10.0/24
set service nat rule 10 outbound-interface eth0
set service nat rule 10 exclude

set service nat rule 20 type masquerade
set service nat rule 20 source address 192.168.40.0/24
set service nat rule 20 outbound-interface eth0
```

For a source NAT rule on eth0:
- rule 10 excludes from the NAT process traffic from the local subnet destined to the remote site(if you have multiple local or remote subnets, add as needed 11, 12, etc. NAT exclusions rules).
- rule 20 is the source NAT type rule that NATs traffic.

```
set service nat rule 10 type source
set service nat rule 10 source address 192.168.40.0/24
set service nat rule 10 destination address 192.168.10.0/24
set service nat rule 10 outbound-interface eth0
set service nat rule 10 exclude
```

```
set service nat rule 20 type source
set service nat rule 20 source address 192.168.40.0/24
set service nat rule 20 outbound-interface eth0
set service nat rule 20 outside-address address 192.168.22.240
```

4.11 Vyatta as web proxy + Vyatta as IPsec tunnel mode VPN gateway

You may enable the web proxy service on Vyatta, and while Vyatta terminates an IPsec tunnel mode s2s connection, you may attempt to access a web server located on the remote site.
Unfortunately due to the kernel route from the s2s connection you may end up with the bellow screen(on a web proxy client)-basically Vyatta will issue ARP requests on its eth0 interface for the needed IP address(in my case 192.168.10.2), and if you just delete the kernel route will simply send the HTTP requests directly over the eth0 interface(due to the default route)-with some extra steps(and having the needed IKE QM proxy ids in place) you may be able to send proxied web traffic to the remote site though-:

```
vyatta@vyatta:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

S>* 0.0.0.0/0 [1/0] via 192.168.22.1, eth0
C>* 127.0.0.0/8 is directly connected, lo
K>* 192.168.10.0/24 is directly connected, eth0
C>* 192.168.22.0/24 is directly connected, eth0
C>* 192.168.40.0/24 is directly connected, eth1
vyatta@vyatta:~$
```

Figure29: Kernel route

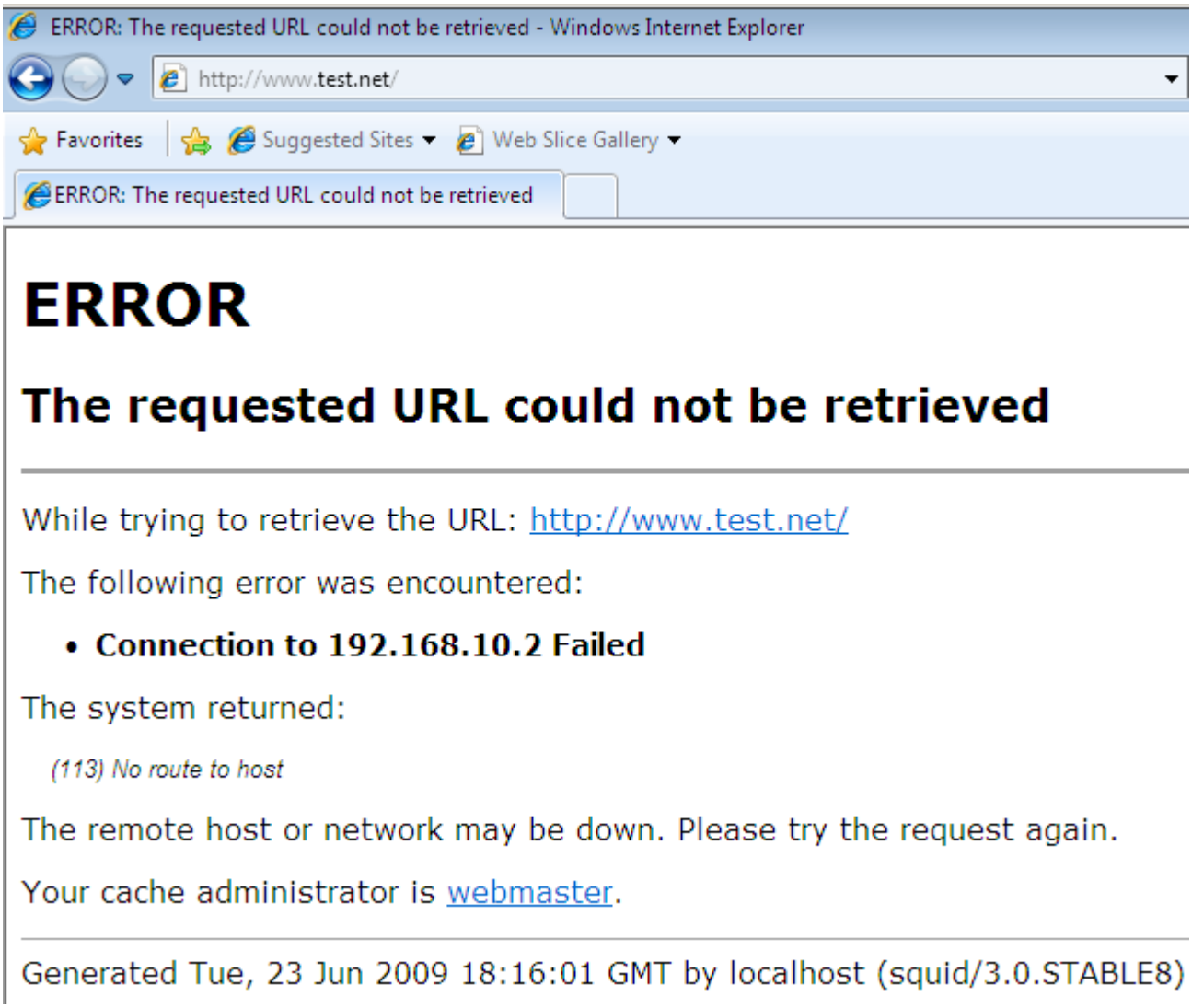

Figure30: Web Proxy Error

4.12 GRE over IPsec: Traffic between local and remote subnets on the tunnel interface(tun1)

We want to filter traffic between local and remote subnet and vice-versa, see **Figure31**.

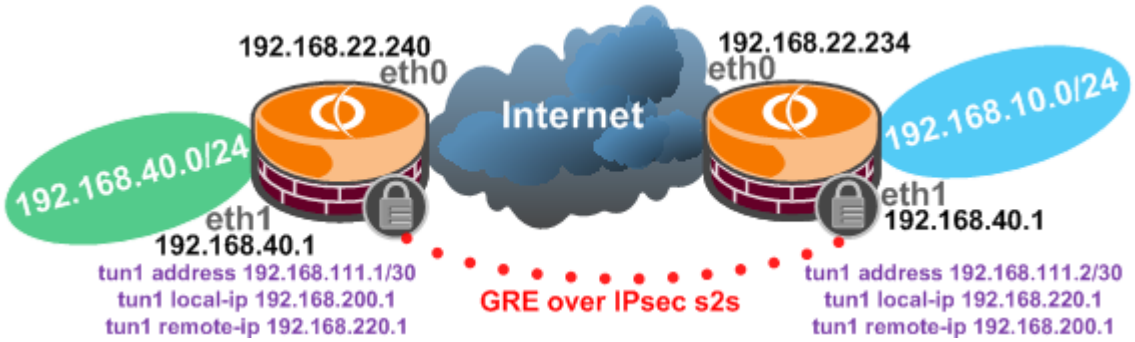

Figure31: Network Diagram

Traffic between local and remote subnets on the tunnel interface(tun1):
We can firewall the traffic between the local and remote subnets by applying **in** and **out** firewall instances on the tunnel interface(tun1).

*** **In** firewall instance on the tunnel interface(tun1).

set firewall name tun1in rule 10 action accept

```
set firewall name tun1in rule 10 source address 192.168.10.0/24
set firewall name tun1in rule 10 destination address 192.168.40.0/24
set firewall name tun1in rule 10 state new enable
set firewall name tun1in rule 10 state established enable
set firewall name tun1in rule 10 state related enable

set interfaces tunnel tun1 firewall in name tun1in
```

*** **Out** firewall instance on the tunnel interface(tun1).

```
set firewall name tun1out rule 10 action accept
set firewall name tun1out rule 10 source address 192.168.40.0/24
set firewall name tun1out rule 10 destination address 192.168.10.0/24
set firewall name tun1out rule 10 state new enable
set firewall name tun1out rule 10 state established enable
set firewall name tun1out rule 10 state related enable

set interfaces tunnel tun1 firewall out name tun1out
```

4.13 GRE over IPsec: Traffic between local and remote subnets on the internal interface(eth1)

We can also partially firewall the traffic between the local and remote subnets by applying **in** and **out** firewall instances on the internal interface(eth1).

*** **In** firewall instance on the internal interface(eth1) ***
- rule 10 allows traffic between the subnet behind eth1 and the remote site.
- rule 20 is just a test rule, allowing web traffic from the subnet behind eth1 destined to the Internet(note that I've excluded the remote subnet, I could not apply the exclusion if the remote site would have contained multiple non-contiguous subnets).
- rule 30 is just another test rule, allowing DNS traffic from the subnet behind eth1 destined to the "ISP DNS server".

```
set firewall name eth1in rule 10 action accept
set firewall name eth1in rule 10 destination address 192.168.10.0/24
set firewall name eth1in rule 10 source address 192.168.40.0/24
set firewall name eth1in rule 10 state new enable
set firewall name eth1in rule 10 state established enable
set firewall name eth1in rule 10 state related enable

set firewall name eth1in rule 20 action accept
set firewall name eth1in rule 20 protocol tcp
set firewall name eth1in rule 20 destination address !192.168.10.0/24
set firewall name eth1in rule 20 destination port 80,443
set firewall name eth1in rule 20 source address 192.168.40.0/24
set firewall name eth1in rule 20 state new enable
set firewall name eth1in rule 20 state established enable
set firewall name eth1in rule 20 state related enable

set firewall name eth1in rule 30 action accept
set firewall name eth1in rule 30 protocol udp
set firewall name eth1in rule 30 destination address 192.168.22.1
set firewall name eth1in rule 30 destination port 53
set firewall name eth1in rule 30 source address 192.168.40.0/24
set firewall name eth1in rule 30 state new enable
```

```
set firewall name eth1in rule 30 state established enable
set firewall name eth1in rule 30 state related enable

set interfaces ethernet eth1 firewall in name eth1in
```

*** **Out** firewall instance on the internal interface(eth1) ***
- rule 10 allows traffic between the subnet behind eth1 and the remote site.
- rule 20 is just a test rule, allowing returning web traffic from the Internet(note that I've excluded the remote subnet, I could not apply the exclusion if the remote site would have contained multiple non-contiguous subnets) destined to the subnet behind eth1.
- rule 30 is just another test rule, allowing returning DNS traffic from the "ISP DNS server" destined to the subnet behind eth1.

```
set firewall name eth1out rule 10 action accept
set firewall name eth1out rule 10 protocol all
set firewall name eth1out rule 10 destination address 192.168.40.0/24
set firewall name eth1out rule 10 source address 192.168.10.0/24
set firewall name eth1out rule 10 state new enable
set firewall name eth1out rule 10 state established enable
set firewall name eth1out rule 10 state related enable

set firewall name eth1out rule 20 action accept
set firewall name eth1out rule 20 protocol udp
set firewall name eth1out rule 20 destination address 192.168.40.0/24
set firewall name eth1out rule 20 source address 192.168.22.1
set firewall name eth1out rule 20 source port 53
set firewall name eth1out rule 20 state established enable
set firewall name eth1out rule 20 state related enable

set firewall name eth1out rule 30 action accept
set firewall name eth1out rule 30 protocol tcp
set firewall name eth1out rule 30 destination address 192.168.40.0/24
set firewall name eth1out rule 30 source address !192.168.10.0/24
set firewall name eth1out rule 30 source port 80,443
set firewall name eth1out rule 30 state established enable
set firewall name eth1out rule 30 state related enable

set interfaces ethernet eth1 firewall out name eth1out
```

4.14 GRE over IPsec: In and out firewall instances on the Internet facing interface(eth0)

The bellow firewall instances on the eth0 interface do not affect our GRE/IPsec s2s(I've exemplified them just for testing).

*** **In** and **Out** firewall instance on the Internet facing interface(eth0) ***

*** **In** firewall instance on the Internet facing interface(eth0) ***
- rule 10 is just a test rule, allowing returning web traffic from the Internet(note that I've excluded the remote subnet, I could not apply the exclusion if the remote site would have contained multiple non-contiguous subnets) destined to the subnet behind eth1.
- rule 20 is just another test rule, allowing returning DNS traffic from the "ISP DNS server" destined to the subnet behind eth1.

```
set firewall name eth0in rule 10 action accept
```

```
set firewall name eth0in rule 10 protocol tcp
set firewall name eth0in rule 10 destination address 192.168.40.0/24
set firewall name eth0in rule 10 source address !192.168.10.0/24
set firewall name eth0in rule 10 source port 80,443
set firewall name eth0in rule 10 state established enable
set firewall name eth0in rule 10 state related enable

set firewall name eth0in rule 20 action accept
set firewall name eth0in rule 20 protocol udp
set firewall name eth0in rule 20 destination address 192.168.40.0/24
set firewall name eth0in rule 20 source address 192.168.22.1
set firewall name eth0in rule 20 source port 53
set firewall name eth0in rule 20 state established enable
set firewall name eth0in rule 20 state related enable

set interfaces ethernet eth0 firewall in name eth0in
```

*** **Out** firewall instance on the Internet facing interface(eth0) ***
- rule 20 is just a test rule, allowing web traffic from the subnet behind eth1 destined to the Internet(note that I've excluded the remote subnet, I could not apply the exclusion if the remote site would have contained multiple non-contiguous subnets).
- rule 30 is just another test rule, allowing DNS traffic from the subnet behind eth1 destined to the "ISP DNS server".

```
set firewall name eth0out rule 10 action accept
set firewall name eth0out rule 10 protocol tcp
set firewall name eth0out rule 10 source address 192.168.40.0/24
set firewall name eth0out rule 10 destination port 80,443
set firewall name eth0out rule 10 destination address !192.168.10.0/24
set firewall name eth0out rule 10 state new enable
set firewall name eth0out rule 10 state established enable
set firewall name eth0out rule 10 state related enable

set firewall name eth0out rule 20 action accept
set firewall name eth0out rule 20 protocol udp
set firewall name eth0out rule 20 source address 192.168.40.0/24
set firewall name eth0out rule 20 destination port 53
set firewall name eth0out rule 20 destination address 192.168.22.1
set firewall name eth0out rule 20 state new enable
set firewall name eth0out rule 20 state established enable
set firewall name eth0out rule 20 state related enable

set interfaces ethernet eth0 firewall out name eth0out
```

5. Publish servers with Vyatta

Say you have some servers behind Vyatta(HTTP server, FTP server, PPTP or L2TP/IPsec VPN servers, etc.), see **Figure32**.
For convenience I've placed them on a "general" internal network, but you can make that a perimeter(DMZ) network if you desire.

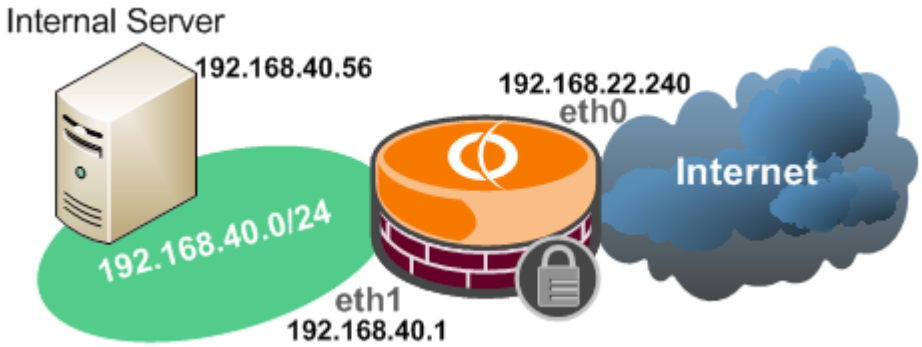

Figure32: Network Diagram

5.1 Publish a web(HTTP) server

Say you have a web server behind Vyatta.

You may have on Vyatta the following NAT rules(we have assumed that the Internet facing interface has a static IP address):

- either a masquerade NAT rule(in case you have a single IP address on the Internet facing interface).

```
set service nat rule 20 type masquerade
set service nat rule 20 source address 192.168.40.0/24
set service nat rule 20 outbound-interface eth0
```

- or a source NAT rule(in case you have a single or multiple IP address(es) on the Internet facing interface).

```
set service nat rule 20 type source
set service nat rule 20 source address 192.168.40.0/24
set service nat rule 20 outbound-interface eth0
set service nat rule 20 outside-address address 192.168.22.240
```

First we need to add a DNAT rule:

```
set service nat rule 10 type destination
set service nat rule 10 protocol tcp
set service nat rule 10 inside-address port 80
set service nat rule 10 inside-address address 192.168.40.56
set service nat rule 10 inbound-interface eth0
set service nat rule 10 destination address 192.168.22.240
set service nat rule 10 destination port 80
```

And then we can add our firewall rules.

```
set firewall conntrack-tcp-loose disable
```

*** **In** firewall instance on the internal interface(eth1) ***

```
set firewall name eth1in rule 10 action accept
set firewall name eth1in rule 10 protocol tcp
set firewall name eth1in rule 10 source address 192.168.40.56
set firewall name eth1in rule 10 source port 80
set firewall name eth1in rule 10 state established enable
set firewall name eth1in rule 10 state related enable
```

```
set interfaces ethernet eth1 firewall in name eth1in
```

*** **Out** firewall instance on the internal interface(eth1) ***

```
set firewall name eth1out rule 10 action accept
set firewall name eth1out rule 10 protocol tcp
set firewall name eth1out rule 10 destination address 192.168.40.56
set firewall name eth1out rule 10 destination port 80
set firewall name eth1out rule 10 state new enable
set firewall name eth1out rule 10 state established enable
set firewall name eth1out rule 10 state related enable

set interfaces ethernet eth1 firewall out name eth1out
```

*** **In** firewall instance on the Internet facing interface(eth0) ***

```
set firewall name eth0in rule 10 action accept
set firewall name eth0in rule 10 protocol tcp
set firewall name eth0in rule 10 destination address 192.168.40.56
set firewall name eth0in rule 10 destination port 80
set firewall name eth0in rule 10 state new enable
set firewall name eth0in rule 10 state established enable
set firewall name eth0in rule 10 state related enable

set interfaces ethernet eth0 firewall in name eth0in
```

*** **Out** firewall instance on the Internet facing interface(eth0) ***

```
set firewall name eth0out rule 10 action accept
set firewall name eth0out rule 10 protocol tcp
set firewall name eth0out rule 10 source address 192.168.40.56
set firewall name eth0out rule 10 source port 80
set firewall name eth0out rule 10 state established enable
set firewall name eth0out rule 10 state related enable

set interfaces ethernet eth0 firewall out name eth0out
```

And to add a form of SYN flooding protection for our web server, we can add to the **in** firewall instance from the Internet facing interface(eth0), basically an external host would not be able to open more than 20 new connections in 2 seconds to our web server. I know this is not pretty as Vyatta will not attempt to SYN proxy, so if one can spoof another host(as it just have to make its spoofed SYN segments reach Vyatta), it may be able to make Vyatta deny access for the real legitimate host:

```
set firewall name eth0in rule 5 action drop
set firewall name eth0in rule 5 protocol tcp
set firewall name eth0in rule 5 destination address 192.168.40.56
set firewall name eth0in rule 5 destination port 80
set firewall name eth0in rule 5 state new enable
set firewall name eth0in rule 5 recent count 20
set firewall name eth0in rule 5 recent time 2
```

5.2 Publish a web(HTTP) server on an alternate port

Say one more time you have a web server behind Vyatta, but instead of publishing on port 80, you want to use an alternate port say TCP port 5000 on the Internet facing interface and then redirect this TCP port 5000 to the "normal" TCP port 80 on the internal web server.

You may have on Vyatta the following NAT rules(we have assumed that the Internet facing interface has a static IP address):

- either a masquerade NAT rule(in case you have a single IP address on the Internet facing interface).

```
set service nat rule 20 type masquerade
set service nat rule 20 source address 192.168.40.0/24
set service nat rule 20 outbound-interface eth0
```

- or a source NAT rule(in case you have a single or multiple IP address(es) on the Internet facing interface).

```
set service nat rule 20 type source
set service nat rule 20 source address 192.168.40.0/24
set service nat rule 20 outbound-interface eth0
set service nat rule 20 outside-address address 192.168.22.240
```

First we need to add the DNAT rule:

```
set service nat rule 10 type destination
set service nat rule 10 protocol tcp
set service nat rule 10 inside-address port 80
set service nat rule 10 inside-address address 192.168.40.56
set service nat rule 10 inbound-interface eth0
set service nat rule 10 destination address 192.168.22.240
set service nat rule 10 destination port 5000
```

And then we can add our firewall rules.
Note that the DNAT process takes place before the firewall one, so for example we just need one firewall rule on the **in** firewall instance on the Internet facing interface, allowing TCP port 80 and not TCP port 5000. Thus, basically, the firewall rules from the above scenario where we published the web server on the standard TCP port 80, will remain untouched.

```
set firewall conntrack-tcp-loose disable
```

*** **In** firewall instance on the internal interface(eth1) ***

```
set firewall name eth1in rule 10 action accept
set firewall name eth1in rule 10 protocol tcp
set firewall name eth1in rule 10 source address 192.168.40.56
set firewall name eth1in rule 10 source port 80
set firewall name eth1in rule 10 state established enable
set firewall name eth1in rule 10 state related enable

set interfaces ethernet eth1 firewall in name eth1in
```

*** **Out** firewall instance on the internal interface(eth1) ***

```
set firewall name eth1out rule 10 action accept
```

```
set firewall name eth1out rule 10 protocol tcp
set firewall name eth1out rule 10 destination address 192.168.40.56
set firewall name eth1out rule 10 destination port 80
set firewall name eth1out rule 10 state new enable
set firewall name eth1out rule 10 state established enable
set firewall name eth1out rule 10 state related enable

set interfaces ethernet eth1 firewall out name eth1out
```

*** **In** firewall instance on the Internet facing interface(eth0) ***

```
set firewall name eth0in rule 10 action accept
set firewall name eth0in rule 10 protocol tcp
set firewall name eth0in rule 10 destination address 192.168.40.56
set firewall name eth0in rule 10 destination port 80
set firewall name eth0in rule 10 state new enable
set firewall name eth0in rule 10 state established enable
set firewall name eth0in rule 10 state related enable

set interfaces ethernet eth0 firewall in name eth0in
```

*** **Out** firewall instance on the Internet facing interface(eth0) ***

```
set firewall name eth0out rule 10 action accept
set firewall name eth0out rule 10 protocol tcp
set firewall name eth0out rule 10 source address 192.168.40.56
set firewall name eth0out rule 10 source port 80
set firewall name eth0out rule 10 state established enable
set firewall name eth0out rule 10 state related enable

set interfaces ethernet eth0 firewall out name eth0out
```

5.3 Publish a FTP server

Say you have a FTP server behind Vyatta.
I will use a Filezilla FTP server.
For convenience, I've specified a pool of port for Passive FTP on this server, see **Figure33**.
Please note that we do not need to specify on the Filezilla FTP server an External Server IP address for passive mode transfers, as the needed FTP modules were already loaded on Vyatta, see **Figure34**.

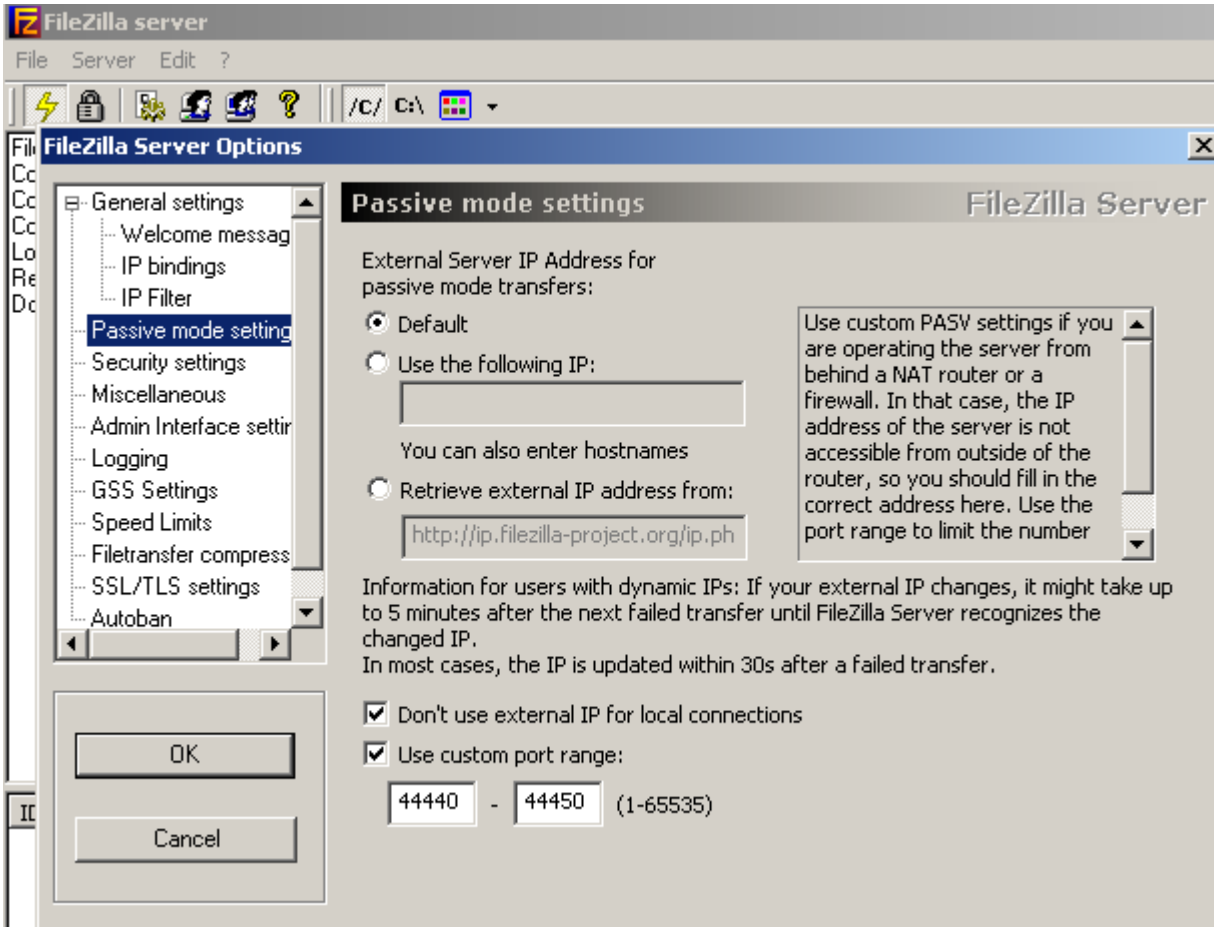

Figure33: Filezilla Passive FTP Settings

```
vyatta@vyatta:~$ zcat /proc/config.gz | grep -i ftp
CONFIG_IP_VS_FTP=m
CONFIG_NF_CONNTRACK_FTP=m
CONFIG_NF_CONNTRACK_TFTP=m
CONFIG_NF_NAT_FTP=m
CONFIG_NF_NAT_TFTP=m
vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$ cat /proc/modules | grep -i ftp
nf_nat_tftp 2432 0 - Live 0xd0bca000
nf_nat_ftp 3712 0 - Live 0xd0bc8000
nf_nat 18072 7 iptable_nat,nf_nat_pptp,nf_nat_h323,nf_nat_sip,nf_nat_proto_gre,n
f_nat_tftp,nf_nat_ftp, Live 0xd0be8000
nf_conntrack_tftp 5140 1 nf_nat_tftp, Live 0xd0b8b000
nf_conntrack_ftp 8356 1 nf_nat_ftp, Live 0xd0bc4000
nf_conntrack 61140 15 iptable_nat,xt_NOTRACK,nf_nat_pptp,nf_conntrack_pptp,nf_co
nntrack_proto_gre,nf_nat_h323,nf_conntrack_h323,nf_nat_sip,nf_conntrack_sip,nf_n
at_tftp,nf_nat_ftp,nf_nat,nf_conntrack_ipv4,nf_conntrack_tftp,nf_conntrack_ftp,
Live 0xd0bf8000
```

Figure34: Loaded FTP modules

You may have on Vyatta the following NAT rules(we have assumed that the Internet facing interface has a static IP address):

- either a masquerade NAT rule(in case you have a single IP address on the Internet facing interface).

```
set service nat rule 20 type masquerade
set service nat rule 20 source address 192.168.40.0/24
set service nat rule 20 outbound-interface eth0
```

- or a source NAT rule(in case you have a single or multiple IP address(es) on the Internet facing interface).

```
set service nat rule 20 type source
set service nat rule 20 source address 192.168.40.0/24
set service nat rule 20 outbound-interface eth0
set service nat rule 20 outside-address address 192.168.22.240
```

First we need to add the DNAT rule, as can be seen I've only translated TCP port 21(FTP control channel), as for the rest the needed modules on Vyatta(see **Figure34**) will take care:

```
set service nat rule 10 type destination
set service nat rule 10 protocol tcp
set service nat rule 10 inside-address port 21
set service nat rule 10 inside-address address 192.168.40.56
set service nat rule 10 inbound-interface eth0
set service nat rule 10 destination address 192.168.22.240
set service nat rule 10 destination port 21
```

Next, we can configure the firewall rules, taking advantage of the modules loaded on Vyatta.
- rule 10 allows the FTP control channel.
- rule 15 allows the FTP data channel for active FTP.
- rule 20 allows the FTP data channel for passive FTP.

```
set firewall conntrack-tcp-loose disable
```

*** **In** firewall instance on the internal interface(eth1) ***

```
set firewall name eth1in rule 10 action accept
set firewall name eth1in rule 10 protocol tcp
set firewall name eth1in rule 10 source address 192.168.40.56
set firewall name eth1in rule 10 source port 21
set firewall name eth1in rule 10 state established enable
set firewall name eth1in rule 10 state related enable

set firewall name eth1in rule 15 action accept
set firewall name eth1in rule 15 protocol tcp
set firewall name eth1in rule 15 source address 192.168.40.56
set firewall name eth1in rule 15 source port 20
set firewall name eth1in rule 15 state established enable
set firewall name eth1in rule 15 state related enable

set firewall name eth1in rule 20 action accept
set firewall name eth1in rule 20 protocol tcp
set firewall name eth1in rule 20 source address 192.168.40.56
set firewall name eth1in rule 20 source port 44440-44450
set firewall name eth1in rule 20 state established enable
set firewall name eth1in rule 20 state related enable

set interfaces ethernet eth1 firewall in name eth1in
```

*** **Out** firewall instance on the internal interface(eth1) ***

```
set firewall name eth1out rule 10 action accept
set firewall name eth1out rule 10 protocol tcp
set firewall name eth1out rule 10 destination address 192.168.40.56
set firewall name eth1out rule 10 destination port 21
set firewall name eth1out rule 10 state new enable
set firewall name eth1out rule 10 state established enable
set firewall name eth1out rule 10 state related enable

set firewall name eth1out rule 15 action accept
set firewall name eth1out rule 15 protocol tcp
set firewall name eth1out rule 15 destination address 192.168.40.56
set firewall name eth1out rule 15 destination port 20
set firewall name eth1out rule 15 state established enable
set firewall name eth1out rule 15 state related enable

set firewall name eth1out rule 20 action accept
set firewall name eth1out rule 20 protocol tcp
set firewall name eth1out rule 20 destination address 192.168.40.56
set firewall name eth1out rule 20 destination port 44440-44450
set firewall name eth1out rule 20 state established enable
set firewall name eth1out rule 20 state related enable

set interfaces ethernet eth1 firewall out name eth1out
```

*** **In** firewall instance on the Internet facing interface(eth0) ***

```
set firewall name eth0in rule 10 action accept
set firewall name eth0in rule 10 protocol tcp
set firewall name eth0in rule 10 destination address 192.168.40.56
set firewall name eth0in rule 10 destination port 21
set firewall name eth0in rule 10 state new enable
set firewall name eth0in rule 10 state established enable
set firewall name eth0in rule 10 state related enable

set firewall name eth0in rule 15 action accept
set firewall name eth0in rule 15 protocol tcp
set firewall name eth0in rule 15 destination address 192.168.40.56
set firewall name eth0in rule 15 destination port 20
set firewall name eth0in rule 15 state established enable
set firewall name eth0in rule 15 state related enable

set firewall name eth0in rule 20 action accept
set firewall name eth0in rule 20 protocol tcp
set firewall name eth0in rule 20 destination address 192.168.40.56
set firewall name eth0in rule 20 destination port 44440-44450
set firewall name eth0in rule 20 state established enable
set firewall name eth0in rule 20 state related enable

set interfaces ethernet eth0 firewall in name eth0in
```

*** **Out** firewall instance on the Internet facing interface(eth0) ***

```
set firewall name eth0out rule 10 action accept
set firewall name eth0out rule 10 protocol tcp
set firewall name eth0out rule 10 source address 192.168.40.56
set firewall name eth0out rule 10 source port 21
set firewall name eth0out rule 10 state established enable
set firewall name eth0out rule 10 state related enable

set firewall name eth0out rule 15 action accept
set firewall name eth0out rule 15 protocol tcp
set firewall name eth0out rule 15 source address 192.168.40.56
set firewall name eth0out rule 15 source port 20
set firewall name eth0out rule 15 state established enable
set firewall name eth0out rule 15 state related enable

set firewall name eth0out rule 20 action accept
set firewall name eth0out rule 20 protocol tcp
set firewall name eth0out rule 20 source address 192.168.40.56
set firewall name eth0out rule 20 source port 44440-44450
set firewall name eth0out rule 20 state established enable
set firewall name eth0out rule 20 state related enable

set interfaces ethernet eth0 firewall out name eth0out
```

And to add a form of SYN flooding protection for our FTP server, we can add to the **in** firewall instance from the Internet facing interface(eth0), basically an external host would not be able to open more than 20 new connections(FTP control channel port TCP 21) in 2 seconds to our FTP server. I know this is not pretty as Vyatta will not attempt to SYN proxy, so if one can spoof another host(as it just have to make its spoofed SYN segments reach Vyatta), it may be able to make Vyatta deny access for the real legitimate host:

```
set firewall name eth0in rule 5 action drop
set firewall name eth0in rule 5 protocol tcp
set firewall name eth0in rule 5 destination address 192.168.40.56
set firewall name eth0in rule 5 destination port 21
set firewall name eth0in rule 5 state new enable
set firewall name eth0in rule 5 recent count 20
set firewall name eth0in rule 5 recent time 2
```

5.4 Publish a FTP server on an alternate port

Say one more time you have a FTP server behind Vyatta.
As above, I will use a Filezilla FTP server, same settings.
Now instead of publishing on port 21, you want to use an alternate port say TCP port 5000 on the Internet facing interface and then redirect this TCP port 5000 to the "normal" TCP port 21 on the internal FTP server.

You may have on Vyatta the following NAT rules(we have assumed that the Internet facing interface has a static IP address):

- either a masquerade NAT rule(in case you have a single IP address on the Internet facing interface).

```
set service nat rule 20 type masquerade
set service nat rule 20 source address 192.168.40.0/24
set service nat rule 20 outbound-interface eth0
```

- or a source NAT rule(in case you have a single or multiple IP address(es) on the Internet facing interface).

```
set service nat rule 20 type source
set service nat rule 20 source address 192.168.40.0/24
set service nat rule 20 outbound-interface eth0
set service nat rule 20 outside-address address 192.168.22.240
```

First we need to add the DNAT rule, as above, we only need to translate the FTP control channel port(5000), as for the rest the needed modules on Vyatta will take care:

```
set service nat rule 10 type destination
set service nat rule 10 protocol tcp
set service nat rule 10 inside-address port 21
set service nat rule 10 inside-address address 192.168.40.56
set service nat rule 10 inbound-interface eth0
set service nat rule 10 destination address 192.168.22.240
set service nat rule 10 destination port 5000
```

And then we can add our firewall rules.

```
set firewall conntrack-tcp-loose disable
```

Note that the DNAT process takes place before the firewall one, so for example we just need one firewall rule on the **in** firewall instance on the Internet facing interface, allowing TCP port 21 and not TCP port 5000. Thus, basically, the firewall rules from the above scenario where we published the FTP server on the standard TCP port 21, will remain untouched.

*** **In** firewall instance on the internal interface(eth1) ***

```
set firewall name eth1in rule 10 action accept
set firewall name eth1in rule 10 protocol tcp
set firewall name eth1in rule 10 source address 192.168.40.56
set firewall name eth1in rule 10 source port 21
set firewall name eth1in rule 10 state established enable
set firewall name eth1in rule 10 state related enable

set firewall name eth1in rule 15 action accept
set firewall name eth1in rule 15 protocol tcp
set firewall name eth1in rule 15 source address 192.168.40.56
set firewall name eth1in rule 15 source port 20
set firewall name eth1in rule 15 state established enable
set firewall name eth1in rule 15 state related enable

set firewall name eth1in rule 20 action accept
set firewall name eth1in rule 20 protocol tcp
set firewall name eth1in rule 20 source address 192.168.40.56
set firewall name eth1in rule 20 source port 44440-44450
set firewall name eth1in rule 20 state established enable
set firewall name eth1in rule 20 state related enable

set interfaces ethernet eth1 firewall in name eth1in
```

*** **Out** firewall instance on the internal interface(eth1) ***

set firewall name eth1out rule 10 action accept
set firewall name eth1out rule 10 protocol tcp
set firewall name eth1out rule 10 destination address 192.168.40.56
set firewall name eth1out rule 10 destination port 21
set firewall name eth1out rule 10 state new enable
set firewall name eth1out rule 10 state established enable
set firewall name eth1out rule 10 state related enable

set firewall name eth1out rule 15 action accept
set firewall name eth1out rule 15 protocol tcp
set firewall name eth1out rule 15 destination address 192.168.40.56
set firewall name eth1out rule 15 destination port 20
set firewall name eth1out rule 15 state established enable
set firewall name eth1out rule 15 state related enable

set firewall name eth1out rule 20 action accept
set firewall name eth1out rule 20 protocol tcp
set firewall name eth1out rule 20 destination address 192.168.40.56
set firewall name eth1out rule 20 destination port 44440-44450
set firewall name eth1out rule 20 state established enable
set firewall name eth1out rule 20 state related enable

set interfaces ethernet eth1 firewall out name eth1out

*** **In** firewall instance on the Internet facing interface(eth0) ***

set firewall name eth0in rule 10 action accept
set firewall name eth0in rule 10 protocol tcp
set firewall name eth0in rule 10 destination address 192.168.40.56
set firewall name eth0in rule 10 destination port 21
set firewall name eth0in rule 10 state new enable
set firewall name eth0in rule 10 state established enable
set firewall name eth0in rule 10 state related enable

set firewall name eth0in rule 15 action accept
set firewall name eth0in rule 15 protocol tcp
set firewall name eth0in rule 15 destination address 192.168.40.56
set firewall name eth0in rule 15 destination port 20
set firewall name eth0in rule 15 state established enable
set firewall name eth0in rule 15 state related enable

set firewall name eth0in rule 20 action accept
set firewall name eth0in rule 20 protocol tcp
set firewall name eth0in rule 20 destination address 192.168.40.56
set firewall name eth0in rule 20 destination port 44440-44450
set firewall name eth0in rule 20 state established enable
set firewall name eth0in rule 20 state related enable

set interfaces ethernet eth0 firewall in name eth0in

*** **Out** firewall instance on the Internet facing interface(eth0) ***

set firewall name eth0out rule 10 action accept

```
set firewall name eth0out rule 10 protocol tcp
set firewall name eth0out rule 10 source address 192.168.40.56
set firewall name eth0out rule 10 source port 21
set firewall name eth0out rule 10 state established enable
set firewall name eth0out rule 10 state related enable

set firewall name eth0out rule 15 action accept
set firewall name eth0out rule 15 protocol tcp
set firewall name eth0out rule 15 source address 192.168.40.56
set firewall name eth0out rule 15 source port 20
set firewall name eth0out rule 15 state established enable
set firewall name eth0out rule 15 state related enable

set firewall name eth0out rule 20 action accept
set firewall name eth0out rule 20 protocol tcp
set firewall name eth0out rule 20 source address 192.168.40.56
set firewall name eth0out rule 20 source port 44440-44450
set firewall name eth0out rule 20 state established enable
set firewall name eth0out rule 20 state related enable

set interfaces ethernet eth0 firewall out name eth0out
```

5.5 Publish a SMTP server

Say you have a SMTP server behind Vyatta.

The NAT rules, the situation may be a little different now as the SMTP server receives and send emails, and you may want to send emails using the same IP address used to receive emails.
For example, there are multiple IP addresses on the Internet facing interface(eth0), and you want to publish a SMTP server using the public IP address 192.168.22.241(receive), and use the same IP address for sending emails from that SMTP server.

For convenience we've put the SMTP server on a "general purpose" internal network behind Vyatta, but if you desire you may put it on a perimeter network(DMZ).

So first we add these NAT rules:
- a DNAT rule for the SMTP server, rule 10.
- a SNAT rule for the SMTP server, rule 15.
- a SNAT rule for the clients behind Vyatta, rule 20, the source IP address is 192.168.22.240.

```
set service nat rule 10 type destination
set service nat rule 10 protocol tcp
set service nat rule 10 inside-address port 25
set service nat rule 10 inside-address address 192.168.40.56
set service nat rule 10 inbound-interface eth0
set service nat rule 10 destination address 192.168.22.241
set service nat rule 10 destination port 25

set service nat rule 15 type source
set service nat rule 15 source address 192.168.40.56
set service nat rule 15 protocol tcp
set service nat rule 15 destination port 25
set service nat rule 15 outbound-interface eth0
set service nat rule 15 outside-address address 192.168.22.241
```

```
set service nat rule 20 type source
set service nat rule 20 source address 192.168.40.0/24
set service nat rule 20 outbound-interface eth0
set service nat rule 20 outside-address address 192.168.22.240
```

And then we can add our firewall rules.

```
set firewall conntrack-tcp-loose disable
```

*** **In** firewall instance on the internal interface(eth1) ***

```
set firewall name eth1in rule 10 action accept
set firewall name eth1in rule 10 protocol tcp
set firewall name eth1in rule 10 source address 192.168.40.56
set firewall name eth1in rule 10 source port 25
set firewall name eth1in rule 10 state established enable
set firewall name eth1in rule 10 state related enable

set firewall name eth1in rule 20 action accept
set firewall name eth1in rule 20 protocol tcp
set firewall name eth1in rule 20 source address 192.168.40.56
set firewall name eth1in rule 20 state new enable
set firewall name eth1in rule 20 destination port 25
set firewall name eth1in rule 20 state established enable
set firewall name eth1in rule 20 state related enable

set interfaces ethernet eth1 firewall in name eth1in
```

*** **Out** firewall instance on the internal interface(eth1) ***

```
set firewall name eth1out rule 10 action accept
set firewall name eth1out rule 10 protocol tcp
set firewall name eth1out rule 10 destination address 192.168.40.56
set firewall name eth1out rule 10 destination port 25
set firewall name eth1out rule 10 state new enable
set firewall name eth1out rule 10 state established enable
set firewall name eth1out rule 10 state related enable

set firewall name eth1out rule 20 action accept
set firewall name eth1out rule 20 protocol tcp
set firewall name eth1out rule 20 destination address 192.168.40.56
set firewall name eth1out rule 20 source port 25
set firewall name eth1out rule 20 state established enable
set firewall name eth1out rule 20 state related enable

set interfaces ethernet eth1 firewall out name eth1out
```

*** **In** firewall instance on the Internet facing interface(eth0) ***

```
set firewall name eth0in rule 10 action accept
set firewall name eth0in rule 10 protocol tcp
set firewall name eth0in rule 10 destination address 192.168.40.56
```

```
set firewall name eth0in rule 10 destination port 25
set firewall name eth0in rule 10 state new enable
set firewall name eth0in rule 10 state established enable
set firewall name eth0in rule 10 state related enable

set firewall name eth0in rule 20 action accept
set firewall name eth0in rule 20 protocol tcp
set firewall name eth0in rule 20 destination address 192.168.40.56
set firewall name eth0in rule 20 source port 25
set firewall name eth0in rule 20 state established enable
set firewall name eth0in rule 20 state related enable

set interfaces ethernet eth0 firewall in name eth0in
```

*** **Out** firewall instance on the Internet facing interface(eth0) ***

```
set firewall name eth0out rule 10 action accept
set firewall name eth0out rule 10 protocol tcp
set firewall name eth0out rule 10 source address 192.168.40.56
set firewall name eth0out rule 10 source port 25
set firewall name eth0out rule 10 state established enable
set firewall name eth0out rule 10 state related enable

set firewall name eth0out rule 20 action accept
set firewall name eth0out rule 20 protocol tcp
set firewall name eth0out rule 20 source address 192.168.40.56
set firewall name eth0out rule 20 destination port 25
set firewall name eth0out rule 20 state new enable
set firewall name eth0out rule 20 state established enable
set firewall name eth0out rule 20 state related enable

set interfaces ethernet eth0 firewall out name eth0out
```

5.6 Publish a PPTP VPN server

Say you have a PPTP VPN server behind Vyatta.
I will use the PPTP VPN server from another Vyatta.

You may have on the front Vyatta the following NAT rules(we have assumed that the Internet facing interface has a static IP address):

- either a masquerade NAT rule(in case you have a single IP address on the Internet facing interface).

```
set service nat rule 20 type masquerade
set service nat rule 20 source address 192.168.40.0/24
set service nat rule 20 outbound-interface eth0
```

- or a source NAT rule(in case you have a single or multiple IP address(es) on the Internet facing interface).

```
set service nat rule 20 type source
set service nat rule 20 source address 192.168.40.0/24
set service nat rule 20 outbound-interface eth0
set service nat rule 20 outside-address address 192.168.22.240
```

First we need to add the DNAT rule, we only need to translate the PPTP TCP port 1723, as for the GRE protocol the needed modules on Vyatta will take care see **Figure35** and **Figure36**:

```
set service nat rule 10 type destination
set service nat rule 10 protocol tcp
set service nat rule 10 inside-address port 1723
set service nat rule 10 inside-address address 192.168.40.56
set service nat rule 10 inbound-interface eth0
set service nat rule 10 destination address 192.168.22.240
set service nat rule 10 destination port 1723
```

```
vyatta@vyatta:~$ zcat /proc/config.gz | grep -i pptp
CONFIG_NF_CONNTRACK_PPTP=m
CONFIG_NF_NAT_PPTP=m
vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$ cat /proc/modules | grep -i pptp
nf_nat_pptp 3840 0 - Live 0xd0c0f000
nf_conntrack_pptp 6916 1 nf_nat_pptp, Live 0xd0bf5000
nf_conntrack_proto_gre 5760 1 nf_conntrack_pptp, Live 0xd0bf2000
nf_nat_proto_gre 3204 1 nf_nat_pptp, Live 0xd0bcd000
nf_nat 18072 7 iptable_nat,nf_nat_pptp,nf_nat_h323,nf_nat_sip,nf_nat_proto_gre,n
f_nat_tftp,nf_nat_ftp, Live 0xd0be9000
nf_conntrack 61140 15 iptable_nat,xt_NOTRACK,nf_nat_pptp,nf_conntrack_pptp,nf_co
nntrack_proto_gre,nf_nat_h323,nf_conntrack_h323,nf_nat_sip,nf_conntrack_sip,nf_n
at_tftp,nf_nat_ftp,nf_nat,nf_conntrack_ipv4,nf_conntrack_tftp,nf_conntrack_ftp,
Live 0xd0bf9000
vyatta@vyatta:~$
```

Figure35: PPTP modules

```
vyatta@vyatta:~$ zcat /proc/config.gz | grep -i gre
CONFIG_NET_IPGRE=m
CONFIG_NET_IPGRE_BROADCAST=y
CONFIG_NF_CT_PROTO_GRE=m
CONFIG_NF_NAT_PROTO_GRE=m
CONFIG_NET_SCH_GRED=m
CONFIG_NET_SCH_INGRESS=m
vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$
vyatta@vyatta:~$ cat /proc/modules | grep -i gre
nf_conntrack_proto_gre 5760 1 nf_conntrack_pptp, Live 0xd0bf2000
nf_nat_proto_gre 3204 1 nf_nat_pptp, Live 0xd0bcd000
nf_nat 18072 7 iptable_nat,nf_nat_pptp,nf_nat_h323,nf_nat_sip,nf_nat_proto_gre,n
f_nat_tftp,nf_nat_ftp, Live 0xd0be9000
nf_conntrack 61140 15 iptable_nat,xt_NOTRACK,nf_nat_pptp,nf_conntrack_pptp,nf_co
nntrack_proto_gre,nf_nat_h323,nf_conntrack_h323,nf_nat_sip,nf_conntrack_sip,nf_n
at_tftp,nf_nat_ftp,nf_nat,nf_conntrack_ipv4,nf_conntrack_tftp,nf_conntrack_ftp,
Live 0xd0bf9000
vyatta@vyatta:~$
```

Figure36: GRE modules

And then we can add our firewall rules(you don't need to allow new GRE connection through Vyatta):
- rule 10 allows traffic on TCP port 1723.
- rule 20 allows GRE.

```
set firewall conntrack-tcp-loose disable
```

*** **In** firewall instance on the internal interface(eth1) ***

```
set firewall name eth1in rule 10 action accept
set firewall name eth1in rule 10 protocol tcp
set firewall name eth1in rule 10 source address 192.168.40.56
set firewall name eth1in rule 10 source port 1723
set firewall name eth1in rule 10 state established enable
set firewall name eth1in rule 10 state related enable

set firewall name eth1in rule 20 action accept
set firewall name eth1in rule 20 protocol gre
set firewall name eth1in rule 20 source address 192.168.40.56
set firewall name eth1in rule 20 state established enable
set firewall name eth1in rule 20 state related enable

set interfaces ethernet eth1 firewall in name eth1in
```

*** **Out** firewall instance on the internal interface(eth1) ***

```
set firewall name eth1out rule 10 action accept
set firewall name eth1out rule 10 protocol tcp
set firewall name eth1out rule 10 destination address 192.168.40.56
set firewall name eth1out rule 10 destination port 1723
set firewall name eth1out rule 10 state new enable
set firewall name eth1out rule 10 state established enable
set firewall name eth1out rule 10 state related enable

set firewall name eth1out rule 20 action accept
set firewall name eth1out rule 20 protocol gre
set firewall name eth1out rule 20 destination address 192.168.40.56
set firewall name eth1out rule 20 state established enable
set firewall name eth1out rule 20 state related enable

set interfaces ethernet eth1 firewall out name eth1out
```

*** **In** firewall instance on the Internet facing interface(eth0) ***

```
set firewall name eth0in rule 10 action accept
set firewall name eth0in rule 10 protocol tcp
set firewall name eth0in rule 10 destination address 192.168.40.56
set firewall name eth0in rule 10 destination port 1723
set firewall name eth0in rule 10 state new enable
set firewall name eth0in rule 10 state established enable
set firewall name eth0in rule 10 state related enable

set firewall name eth0in rule 20 action accept
set firewall name eth0in rule 20 protocol gre
set firewall name eth0in rule 20 destination address 192.168.40.56
set firewall name eth0in rule 20 state established enable
set firewall name eth0in rule 20 state related enable

set interfaces ethernet eth0 firewall in name eth0in
```

*** **Out** firewall instance on the Internet facing interface(eth0) ***

```
set firewall name eth0out rule 10 action accept
set firewall name eth0out rule 10 protocol tcp
set firewall name eth0out rule 10 source address 192.168.40.56
set firewall name eth0out rule 10 source port 1723
set firewall name eth0out rule 10 state established enable
set firewall name eth0out rule 10 state related enable

set firewall name eth0out rule 20 action accept
set firewall name eth0out rule 20 protocol gre
set firewall name eth0out rule 20 source address 192.168.40.56
set firewall name eth0out rule 20 state established enable
set firewall name eth0out rule 20 state related enable

set interfaces ethernet eth0 firewall out name eth0out
```

5.7 Publish a L2TP/IPsec or "pure" IPsec VPN server behind Vyatta

Say you have an L2TP/IPsec or "pure" IPsec VPN server behind Vyatta.

You may have on Vyatta the following NAT rules(we have assumed that the Internet facing interface has a static IP address):

- either a masquerade NAT rule(in case you have a single IP address on the Internet facing interface).

```
set service nat rule 30 type masquerade
set service nat rule 30 source address 192.168.40.0/24
set service nat rule 30 outbound-interface eth0
```

- or a source NAT rule(in case you have a single or multiple IP address(es) on the Internet facing interface).

```
set service nat rule 30 type source
set service nat rule 30 source address 192.168.40.0/24
set service nat rule 30 outbound-interface eth0
set service nat rule 30 outside-address address 192.168.22.240
```

First we need to add the DNAT rules, we need to translate UDP ports 500 and 4500, as the VPN server is behind a NAT device and we need to allow NAT-T:

```
set service nat rule 10 type destination
set service nat rule 10 protocol udp
set service nat rule 10 inside-address port 500
set service nat rule 10 inside-address address 192.168.40.56
set service nat rule 10 inbound-interface eth0
set service nat rule 10 destination address 192.168.22.240
set service nat rule 10 destination port 500

set service nat rule 20 type destination
set service nat rule 20 protocol udp
set service nat rule 20 inside-address port 4500
set service nat rule 20 inside-address address 192.168.40.56
set service nat rule 20 inbound-interface eth0
```

```
set service nat rule 20 destination address 192.168.22.240
set service nat rule 20 destination port 4500
```

Next we can create our firewal rules.

*** **In** firewall instance on the internal interface(eth1) ***

```
set firewall name eth1in rule 10 action accept
set firewall name eth1in rule 10 protocol udp
set firewall name eth1in rule 10 source address 192.168.40.56
set firewall name eth1in rule 10 source port 500
set firewall name eth1in rule 10 state established enable
set firewall name eth1in rule 10 state related enable

set firewall name eth1in rule 20 action accept
set firewall name eth1in rule 20 protocol udp
set firewall name eth1in rule 20 source address 192.168.40.56
set firewall name eth1in rule 20 source port 4500
set firewall name eth1in rule 20 state established enable
set firewall name eth1in rule 20 state related enable

set interfaces ethernet eth1 firewall in name eth1in
```

*** **Out** firewall instance on the internal interface(eth1) ***

```
set firewall name eth1out rule 10 action accept
set firewall name eth1out rule 10 protocol udp
set firewall name eth1out rule 10 destination address 192.168.40.56
set firewall name eth1out rule 10 destination port 500
set firewall name eth1out rule 10 state new enable
set firewall name eth1out rule 10 state established enable
set firewall name eth1out rule 10 state related enable

set firewall name eth1out rule 20 action accept
set firewall name eth1out rule 20 protocol udp
set firewall name eth1out rule 20 destination address 192.168.40.56
set firewall name eth1out rule 20 destination port 4500
set firewall name eth1out rule 20 state new enable
set firewall name eth1out rule 20 state established enable
set firewall name eth1out rule 20 state related enable

set interfaces ethernet eth1 firewall out name eth1out
```

*** **In** firewall instance on the Internet facing interface(eth0) ***

```
set firewall name eth0in rule 10 action accept
set firewall name eth0in rule 10 protocol udp
set firewall name eth0in rule 10 destination address 192.168.40.56
set firewall name eth0in rule 10 destination port 500
set firewall name eth0in rule 10 state new enable
set firewall name eth0in rule 10 state established enable
set firewall name eth0in rule 10 state related enable
```

```
set firewall name eth0in rule 20 action accept
set firewall name eth0in rule 20 protocol udp
set firewall name eth0in rule 20 destination address 192.168.40.56
set firewall name eth0in rule 20 destination port 4500
set firewall name eth0in rule 20 state new enable
set firewall name eth0in rule 20 state established enable
set firewall name eth0in rule 20 state related enable

set interfaces ethernet eth0 firewall in name eth0in
```

*** **Out** firewall instance on the Internet facing interface(eth0) ***

```
set firewall name eth0out rule 10 action accept
set firewall name eth0out rule 10 protocol udp
set firewall name eth0out rule 10 source address 192.168.40.56
set firewall name eth0out rule 10 source port 500
set firewall name eth0out rule 10 state established enable
set firewall name eth0out rule 10 state related enable

set firewall name eth0out rule 20 action accept
set firewall name eth0out rule 20 protocol udp
set firewall name eth0out rule 20 source address 192.168.40.56
set firewall name eth0out rule 20 source port 4500
set firewall name eth0out rule 20 state established enable
set firewall name eth0out rule 20 state related enable

set interfaces ethernet eth0 firewall out name eth0out
```

6. Small home router behavior

Please note that this is far from being the most secure configuration.
It rather "mirrors" the configuration of a typical home router.
Not very experimented users or people needing this type of behavior may configure the firewall like so.
If one desires more protection, it can use the firewall examples from above to have granular firewall rules.
One difference from a typical home router may be that, although some of such devices are supposed to incorporate a "stateful packet inspection firewall", in practice this may quite not be the case.

The diagram, see **Figure37**.

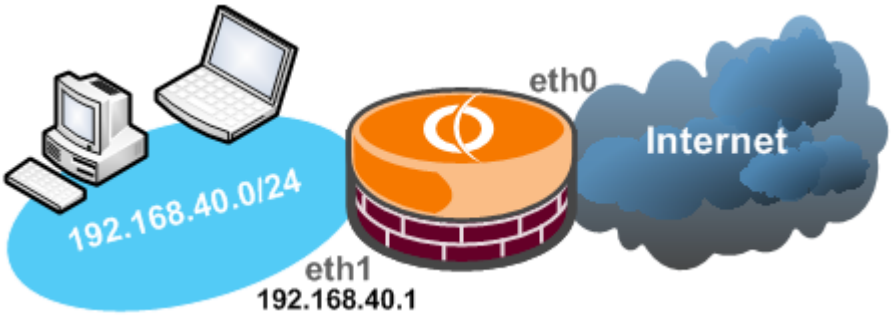

Figure37: Network Diagram

We will have on Vyatta:
- the Internet facing interface(eth0) will receive its IP config from DHCP.
- the internal subnet will be 192.168.40.0/24.
- for management we will enable SSH and the web GUI. They will only be accessible from the internal

network.
- we will use a masquerade type NAT rule on the the Internet facing interface(eth0).
- we will enable the DHCP server on the internal interface(eth1) to serve the internal clients.
- we will enable the DNS forwarding service, internal clients will use Vyatta(the IP address from the internal interface eth1) as their DNS server, and Vyatta will forward their queries to the DNS servers received from the ISP through DHCP.
- stateful packet inspection will be on to protect your internal network.

Configure the ethernet interfaces:

```
set interfaces ethernet eth0 address dhcp
set interfaces ethernet eth1 address 192.168.40.0/24
commit
```

Configure the NAT masquerade type rule:

```
set service nat rule 10 type masquerade
set service nat rule 10 source address 192.168.40.0/24
set service nat rule 10 outbound-interface eth0
commit
```

Enable SSH and the web GUI:

```
set service ssh protocol-version 2
set service https
commit
```

Enable the DHCP server:

```
set service dhcp-server shared-network-name my authoritative enable
set service dhcp-server shared-network-name my subnet 192.168.40.0/24 start 192.168.40.150 stop
192.168.40.200
set service dhcp-server shared-network-name my subnet 192.168.40.0/24 dns-server 192.168.40.1
set service dhcp-server shared-network-name my subnet 192.168.40.0/24 default-router 192.168.40.1
set service dhcp-server shared-network-name my subnet 192.168.40.0/24 lease 86400
commit
```

Enable the DNS forwarding:

```
set service dns forwarding listen-on eth1
set service dns forwarding dhcp eth0
commit
```

Save the configuration:

```
save
```

And now you should check if things work as they are supposed, you have internet connectivity, etc.

If things work fine, you may proceed and configure the firewall -obviously you can configure the firewall before you connect the Vyatta machine to the Internet, if you don't want to do that before the firewall was configured-(as already said, you can practice in a virtual lab, say using VMware Server).

First I will enable the syn-cookies and disable the conntrack-tcp-loose.

```
set firewall syn-cookies enable
set firewall conntrack-tcp-loose disable
```

*** **Local** firewall instance on the internal interface(eth1) ***
Since for a home user may be hard to figure it out exactly what it needs, basically I will allow everything from the internal network(if you don't want that, you can allow the needed services on Vyatta as we discussed before).

```
set firewall name eth1local rule 10 action accept
set firewall name eth1local rule 10 source address 192.168.40.0/24
set firewall name eth1local rule 10 state new enable
set firewall name eth1local rule 10 state established enable
set firewall name eth1local rule 10 state related enable

set interfaces ethernet eth1 firewall local name eth1local
```

*** **In** firewall instance on the internal interface(eth1) ***
Basically I will allow everything from the internal network(we have a basic anti-spoofing mechanism since we filter based on source address).

```
set firewall name eth1in rule 10 action accept
set firewall name eth1in rule 10 source address 192.168.40.0/24
set firewall name eth1in rule 10 state new enable
set firewall name eth1in rule 10 state established enable
set firewall name eth1in rule 10 state related enable

set interfaces ethernet eth1 firewall in name eth1in
```

*** **Local** firewall instance on the Internet facing interface(eth0) ***
As we already saw, Vyatta itself needs access to some service, like DNS name resolution, NTP, HTTP for updates, etc.
Since for a home user may be hard to figure it out exactly what it needs, we will permit on the Internet facing interface(eth0) only established connections destined to Vyatta itself, so Vyatta will not respond to ping from the Internet, and its ports will come as filtered if one probes for open ports.

```
set firewall name eth0local rule 10 action accept
set firewall name eth0local rule 10 state established enable
set firewall name eth0local rule 10 state related enable

set interfaces ethernet eth0 firewall local name eth0local
```

*** **In** firewall instance on the Internet facing interface(eth0) ***
We will permit on the Internet facing interface(eth0) only established connections(initiated from the internal network, we have a basic anti-spoofing mechanism since we filter based on destination address), so new connections cannot pass through the eth0 interface destined to the internal network.

```
set firewall name eth0in rule 10 action accept
set firewall name eth0in rule 10 destination address 192.168.40.0/24
set firewall name eth0in rule 10 state established enable
set firewall name eth0in rule 10 state related enable
```

```
set interfaces ethernet eth0 firewall in name eth0in
```

Commit and save the configuration:

```
commit
save
```

And that's it, we don't necessarily need **out** firewall instances, as things are pretty clear as we have only two interfaces.

7. Quickly display firewall rules and view firewall statistics

You may want to quickly display the configured firewall rules. From the **Configuration mode** you may try to display the config using the **show firewall name** command and if you have multiple rules, it may not be the best way from the CLI to have a brief general view of the configured rules, see **Figure38**.
Instead you may use the **Operational mode** and the **show firewall** command, see **Figure39**.
Actually you don't have to leave the **Configuration mode** to run **Operational mode** commands, just use from the **Configuration mode** the **run** parameter, see **Figure40**.

```
vyatta@vyatta# show firewall name eth1in
 rule 10 {
     action accept
     destination {
         address 192.168.22.1
         port 53
     }
     protocol udp
     source {
         address 192.168.40.0/24
     }
     state {
         established enable
         new enable
         related enable
     }
 }
 rule 20 {
     action accept
     destination {
         port 80,443
     }
     protocol tcp
     source {
```

Figure38: Configuration mode - run show firewall eth1in

```
vyatta@vyatta:~$ show firewall eth1in

Active on (eth1,IN)

State Codes: E - Established, I - Invalid, N - New, R - Related

rule  action  source              destination         proto  state
----  ------  ------              -----------         -----  -----
10    ACCEPT  192.168.40.0/24     192.168.22.1        udp    E,N,R
                                  dst ports: 53
20    ACCEPT  192.168.40.0/24     0.0.0.0/0           tcp    E,N,R
                                  dst ports: 80,443
1025  DROP    0.0.0.0/0           0.0.0.0/0           all    any
```

Figure39: Operational mode - show firewall eth1in

```
vyatta@vyatta# run show firewall eth1in

Active on (eth1,IN)

State Codes: E - Established, I - Invalid, N - New, R - Related

rule  action  source              destination         proto  state
----  ------  ------              -----------         -----  -----
10    ACCEPT  192.168.40.0/24     192.168.22.1        udp    E,N,R
                                  dst ports: 53
20    ACCEPT  192.168.40.0/24     0.0.0.0/0           tcp    E,N,R
                                  dst ports: 80,443
1025  DROP    0.0.0.0/0           0.0.0.0/0           all    any
```

Figure40: Configuration mode - run show firewall eth1in

Or you may want to view some statistics for your firewall rules, as they may help you see if your firewall rules are being applied(as you may have noticed I've never enabled logging on firewall rules within this article).
You cay use the **Operational mode** and the **show firewall ... statistics** command, see **Figure41**.
As above you don't have to leave the **Configuration mode** to run **Operational mode** commands, just use from the **Configuration mode** the **run** parameter, see **Figure42**.

```
vyatta@vyatta:~$ show firewall eth1in statistics

Active on (eth1,IN)

rule  packets   bytes     action  source              destination
----  -------   -----     ------  ------              -----------
10    59        3702      ACCEPT  192.168.40.0/24     192.168.22.1
20    1312      211K      ACCEPT  192.168.40.0/24     0.0.0.0/0
1025  1         152       DROP    0.0.0.0/0           0.0.0.0/0
```

Figure41: Operational mode - show firewall eth1in statistics

```
vyatta@vyatta# run show firewall eth1in statistics

Active on (eth1,IN)

rule  packets   bytes     action  source              destination
----  -------   -----     ------  ------              -----------
10    59        3702      ACCEPT  192.168.40.0/24     192.168.22.1
20    1312      211K      ACCEPT  192.168.40.0/24     0.0.0.0/0
1025  1         152       DROP    0.0.0.0/0           0.0.0.0/0

[edit]
```

Figure42: Configuration mode - run show firewall eth1in statistics