INTRODUCTION TO FIREWALL DEPLOYMENT

VYOS

Linux (VyOS) Firewall Deployment

After the first boot, load the default configuration and reboot:

```
sudo cp /opt/vyatta/etc/config.boot.default /config/config.boot
reboot
```

Check network interface names: `ip addr`

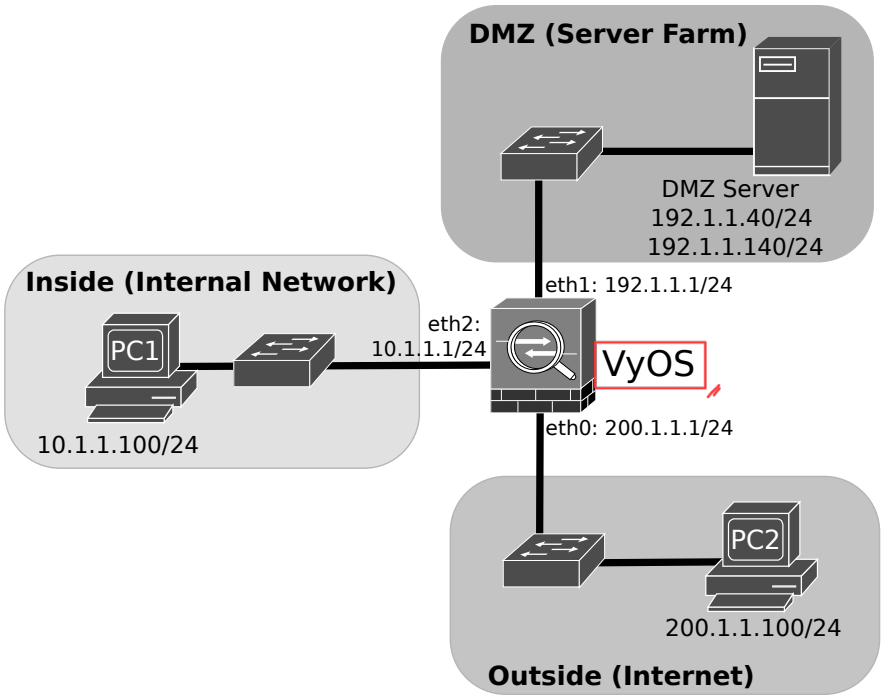
To change the keyboard layout: `set console keymap`

For QEMU GNS3 template use the following parameters: RAM: 512M, Console type: telnet (or none with auto start console checked), HDD Disk interface: ide, Network Adapters: 6, Network Name format: eth{0}.

For VirtualBox GNS3 template use the following parameters: RAM: 512M, Console type: telnet (or none with auto start console checked), Network Adapters: 6, Network Name format: eth{0}, check Network option “Allow GNS3 to use any ... adapter”.

VyOS user guide: https://docs.vyos.io/en/latest/

1. Configure the network depicted in the following figure using GNS3 with PC1 and PC2 as VPCS, the DMZ server as a QEMU Linux server, and the VyOS firewall as a QEMU VM. Configure PCs and Server addresses and gateways.



2. Configure the firewall IPv4 addresses using the following commands.

Enter into configuration mode:

```
$ configure
```

Configure the interfaces IPv4 addresses, commit the configurations and exit the configuration mode:

```
# set interfaces ethernet eth0 address 200.1.1.1/24
# set interfaces ethernet eth1 address 192.1.1.1/24
# set interfaces ethernet eth2 address 10.1.1.1/24
# commit
# exit
```

>> Verify the configured addresses with: `$ show interfaces`

>> Test the full connectivity between all network equipment.

If working as expect save the configuration:

```
$ configure
# save
```

Note: the firewall, by default, has a blank configuration so it allows all traffic and performs all routing mechanisms.

Note2: The “$” prompt denotes the standard/bash mode and the “#” denotes the configuration mode.

3. Configure the firewall NAT/PAT mechanisms. Assume that the network will use the IPv4 public address 192.1.0.1 to 192.1.0.10:

```
# set nat source rule 100 outbound-interface eth0
# set nat source rule 100 source address 10.1.1.0/24
# set nat source rule 100 translation address 192.1.0.1-192.1.0.10
```

>> Use the following command to verify the configured NAT rules: `$ show nat source rules`

>> Start a capture on the link between the firewall (eth0) and the OUTSIDE switch. Ping PC2 from PC1 and verify the correct translation of the source IPv4 addresses.

>> Use the following command to verify the active NAT translations: `$ show nat source translations`

4. Define the network security zones:

```
# set zone-policy zone INSIDE description "Inside (Internal Network)"
# set zone-policy zone INSIDE interface eth2
# set zone-policy zone DMZ description "DMZ (Server Farm)"
# set zone-policy zone DMZ interface eth1
# set zone-policy zone OUTSIDE description "Outside (Internet)"
# set zone-policy zone OUTSIDE interface eth0
# commit
```

To verify the zone policies and firewall rules use the following commands in configuration and standard modes:

```
#/$ show zone-policy
#/$ show firewall
```

>> Test the full (or lack of) connectivity between all network equipment (and IPv4 addresses).

5. Configure the firewalls chains and rules to allow the Inside equipment to ping all Outside devices:

```
# set firewall name FROM-INSIDE-TO-OUTSIDE rule 10 description "Accept ICMP Echo Request"
# set firewall name FROM-INSIDE-TO-OUTSIDE rule 10 action accept
# set firewall name FROM-INSIDE-TO-OUTSIDE rule 10 protocol icmp
# set firewall name FROM-INSIDE-TO-OUTSIDE rule 10 icmp type 8
# set firewall name TO-INSIDE rule 10 description "Accept Established-Related Connections"
# set firewall name TO-INSIDE rule 10 action accept
# set firewall name TO-INSIDE rule 10 state established enable
# set firewall name TO-INSIDE rule 10 state related enable
# set zone-policy zone INSIDE from OUTSIDE firewall name TO-INSIDE
# set zone-policy zone OUTSIDE from INSIDE firewall name FROM-INSIDE-TO-OUTSIDE
# commit
```

Verify the correct configuration in configuration and standard modes:

```
#/$ show zone-policy
#/$ show firewall
```

>> Test the implemented rules, pinging the Server and PC2 from PC1.

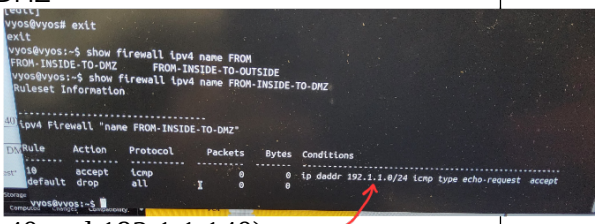
6. Configure the firewalls chains and rules to allow the Inside devices to ping all DMZ (network 192.1.1.0/24) devices:

```
# set firewall name FROM-INSIDE-TO-DMZ rule 10 description "Accept ICMP Echo Request"
# set firewall name FROM-INSIDE-TO-DMZ rule 10 action accept
# set firewall name FROM-INSIDE-TO-DMZ rule 10 protocol icmp
# set firewall name FROM-INSIDE-TO-DMZ rule 10 icmp type 8
# set firewall name FROM-INSIDE-TO-DMZ rule 10 destination address 192.1.1.0/24
# set zone-policy zone INSIDE from DMZ firewall name TO-INSIDE
# set zone-policy zone DMZ from INSIDE firewall name FROM-INSIDE-TO-DMZ
# commit
```

Note: The chain TO-INSIDE was already defined before.

Verify the correct configuration in configuration and standard modes:

```
#/$ show zone-policy
#/$ show firewall
```



>> Test the implemented rules, pinging from PC1 the Server (192.1.1.40 and 192.1.1.140).

7. Configure the firewalls chains and rules to allow the Outside devices to ping the DMZ Server (only IP address 192.1.1.40):

```
# set firewall name FROM-OUTSIDE-TO-DMZ rule 10 description "Accept ICMP Echo Request"
# set firewall name FROM-OUTSIDE-TO-DMZ rule 10 action accept
# set firewall name FROM-OUTSIDE-TO-DMZ rule 10 protocol icmp
# set firewall name FROM-OUTSIDE-TO-DMZ rule 10 icmp type 8
# set firewall name FROM-OUTSIDE-TO-DMZ rule 10 destination address 192.1.1.40
# set firewall name FROM-DMZ-TO-OUTSIDE rule 10 description "Accept Established-Related Connections"
# set firewall name FROM-DMZ-TO-OUTSIDE rule 10 action accept
# set firewall name FROM-DMZ-TO-OUTSIDE rule 10 state established enable
# set firewall name FROM-DMZ-TO-OUTSIDE rule 10 state related enable
# set zone-policy zone OUTSIDE from DMZ firewall name FROM-DMZ-TO-OUTSIDE
# set zone-policy zone DMZ from OUTSIDE firewall name FROM-OUTSIDE-TO-DMZ
# commit
```

Verify the correct configuration in configuration and standard modes:

```
#/$ show zone-policy
#/$ show firewall
```

>> Test the implemented rules, pinging from the PC2 the Server (192.1.1.40 and 192.1.1.140).

8. Add a new rule to the chain FROM-OUTSIDE-TO-DMZ to allow the Outside devices to send UDP packets to port 8080 to the DMZ Server (only IP address 192.1.1.140):

```
# set firewall name FROM-OUTSIDE-TO-DMZ rule 12 description "Accept UDP-8080"
# set firewall name FROM-OUTSIDE-TO-DMZ rule 12 action accept
# set firewall name FROM-OUTSIDE-TO-DMZ rule 12 protocol udp
# set firewall name FROM-OUTSIDE-TO-DMZ rule 12 destination address 192.1.1.140
# set firewall name FROM-OUTSIDE-TO-DMZ rule 12 destination port 8080
# commit
```

Verify the correct configuration in configuration and standard modes:

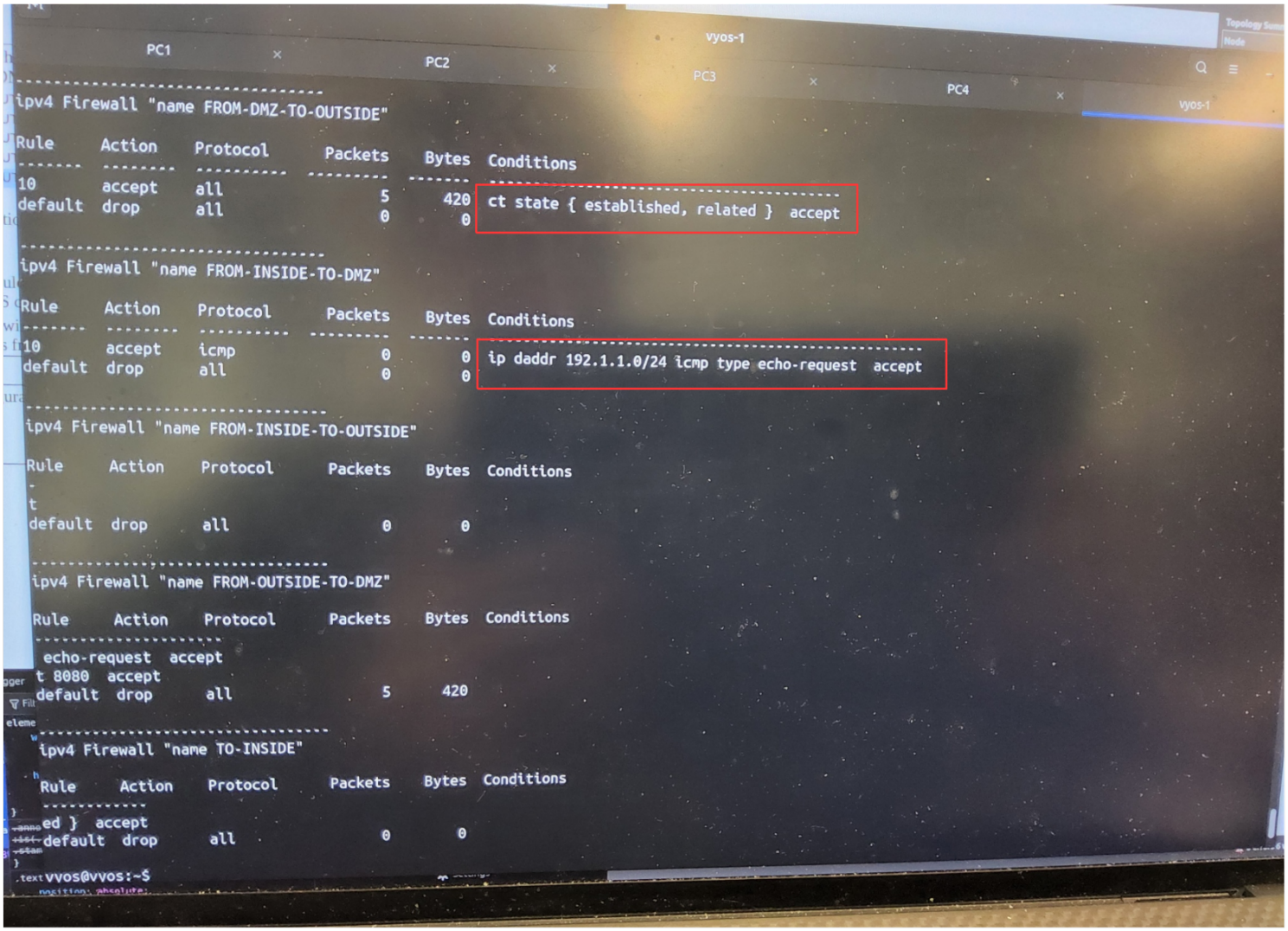
```
#/$ show zone-policy
#/$ show firewall
```

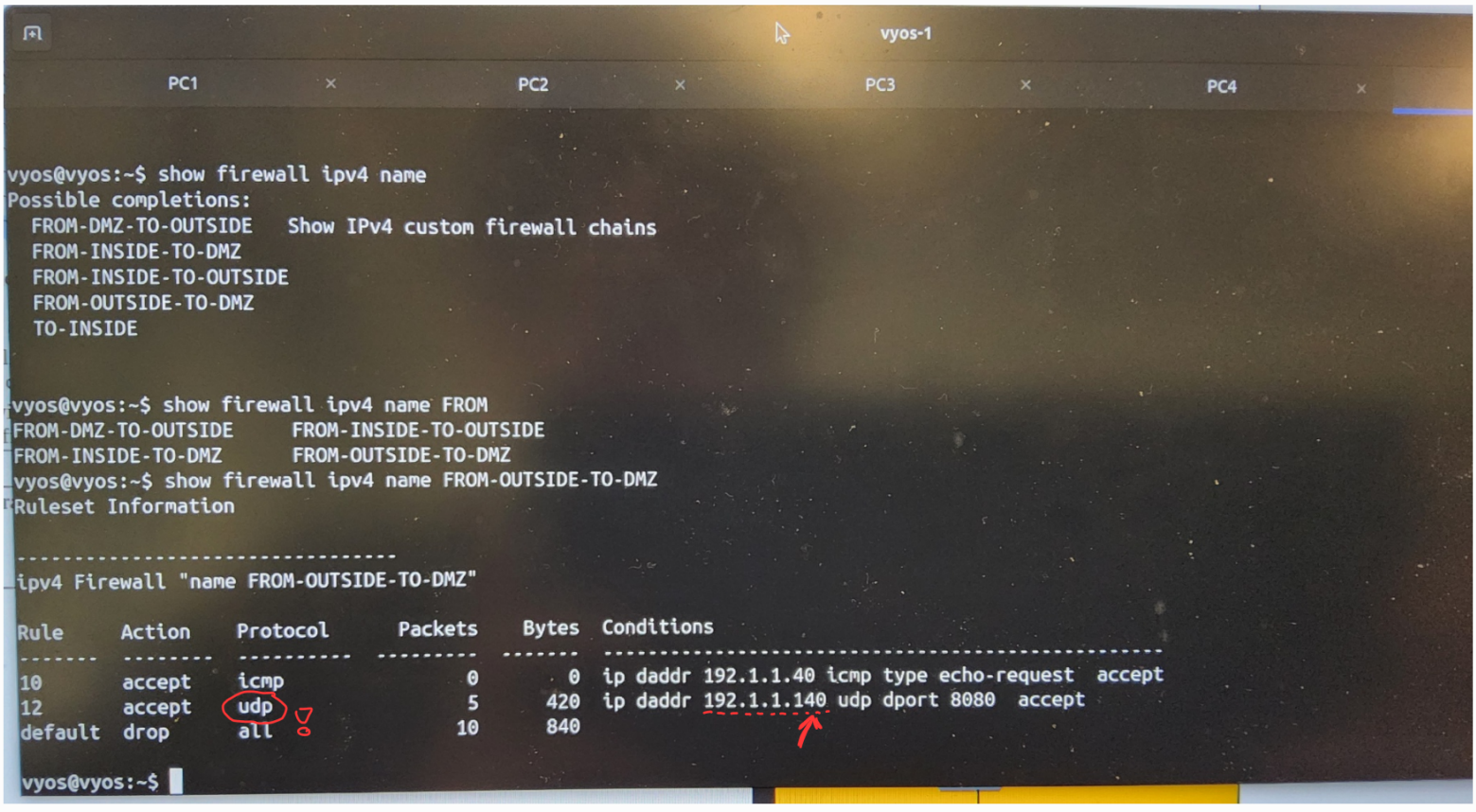
>> Test the implemented rules, pinging with UDP to port 8080 from the PC2 the Server (192.1.1.40 and 192.1.1.140). Use the VPCS command: `ping 192.1.1.140 -P 17 -p 8080`

>> Test the connectivity with the other server IPv4 address, with other UDP ports and test also TCP connections. For TCP pings from the VPCS use command: `ping 192.1.1.140 -P 6 -p 8080`

9. Exit the firewall configuration mode (exit) and analyze the underlying IPTables chains/rules that were created:

```
$ sudo iptables -L
$ sudo iptables -L -t nat
```




vyos-1
PC1
PC2
PC3
PC4
vyos@vyos:~$ show firewall ipv4 name
Possible completions:
FROM-DMZ-TO-OUTSIDE Show IPv4 custom firewall chains
FROM-INSIDE-TO-DMZ
FROM-INSIDE-TO-OUTSIDE
FROM-OUTSIDE-TO-DMZ
TO-INSIDE
vyos@vyos:~$ show firewall ipv4 name FROM
FROM-DMZ-TO-OUTSIDE FROM-INSIDE-TO-OUTSIDE
FROM-INSIDE-TO-DMZ FROM-OUTSIDE-TO-DMZ
vyos@vyos:~$ show firewall ipv4 name FROM-OUTSIDE-TO-DMZ
Ruleset Information
ipv4 Firewall "name FROM-OUTSIDE-TO-DMZ"
Rule Action Protocol Packets Bytes Conditions
10 accept icmp 0 0 ip daddr 192.1.1.40 icmp type echo-request accept
12 accept udp 5 420 ip daddr 192.1.1.140 udp dport 8080 accept
default drop all 10 840
vyos@vyos:~$