Lab 2: Vyatta Firewall and Snort IDS

Aim

The aim of this lab is to build on the basic Vyatta firewall configuration, adding firewalling, IDS, and other hardening capabilities.

Time to Complete:

4 hours (two supervise hours in the lab, and two additional unsupervised hours).

Activities:

- **Complete Lab 2:** Vyatta Firewall and Snort
- **Complete software lab.**

Learning activities:

At the end of this lab, you should understand:

- How to use your own credentials to access the vSoC Cloud.
- How to remotely configure a Vyatta firewall for zones, and set up the firewalling.

Reflective statements (end-of-exercise):

What are the most important things when setting up a host, in order that it can connect with other networks?

Lab 2: Vyatta Firewall and Snort IDS

A Setting up the network

Figure 1 outlines the setup of the lab for routing, where we will assign three network addresses. Again, Interfaces which are connected to the Vyatta firewall will be able to route, but we have to use NAT to allow the DMZ and private networks to connect to the public network.

Our first task is to route through the Vyatta firewall to connect two networks. In the lab you will be assigned two networks in the form:

10.10.*x*.0/24 10.10.*y*.0/24

Demo: https://youtu.be/SJwlt55f_UU

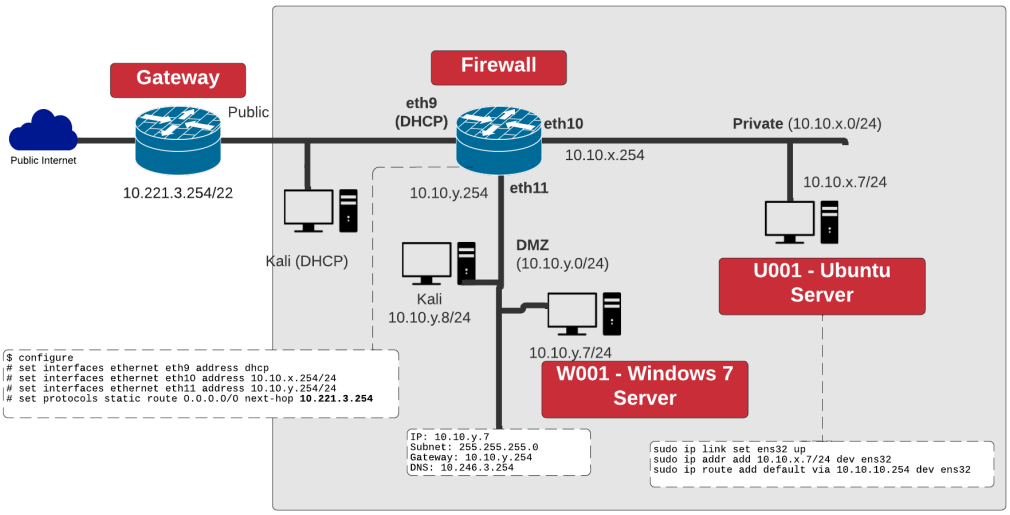

Figure 1: Lab setup (eth0 – Public, eth1 – Private, eth2 – DMZ)

Draw your own network diagram here, by filling-in the blank boxes, with the allocated networks, subnets, and IP addresses:

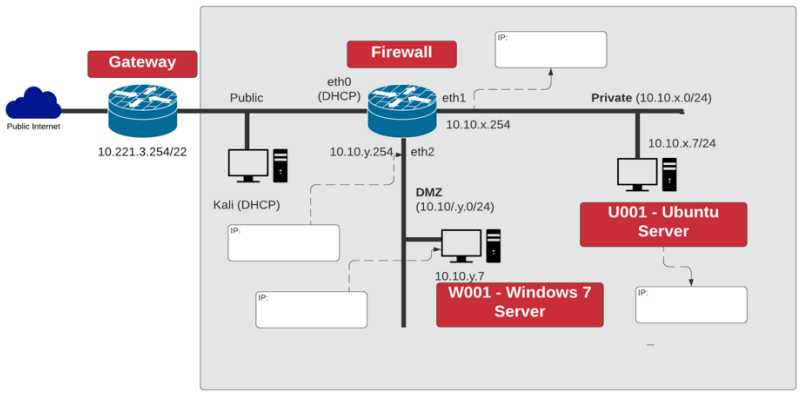

Figure 2: Your network setup (Note: **Gateway address is 10.221.3.254**)

B Configure Router/Firewall for Remote Administration

Note: We are going to use the **Kali host on the DMZ** to configuration the firewall, and use an SSH connection.

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

reboot
```

We typically do not use the console terminal of a firewall for configuration. In the following we will enable one port on the firewall, and then configure it through a remote connection. First configure your Vyatta firewall networking with the following:

```
$ configure
# set interfaces ethernet eth2 address 10.10.y.254/24
```

and then start the Ssh server on the Vyatta firewall:

```
# set service ssh
```

Check the configuration using:

```
# show
# show interfaces
# show service
```

If everything is correct commit the changes, and review the configuration:

```
# commit
# exit
$ show configuration
```

Now setup your **Kali host on the DMZ** for networking on the same private network, and so it will be able to connect to the Vyatta firewall using remote admin with the SSH service:

```
sudo ip link set eth0 up
sudo ip addr add 10.10.y.8/24 dev eth0
sudo ip route add default via 10.10.y.254 dev eth0
```

Now from Kali, check the connectivity using ping to your local connection and the gateway:

```
Can you ping them: [Yes] [No]
```

C Configuring the firewall from Kali

Now we will configure the firewall by creating a file, and copying-and-pasting the config from the file to the firewall via a remote admin session with Telnet.

First, download the following config:

`https://asecuritysite.com/vpart01.txt`

Note you need to use the VMRC console for copy-and-paste to work. Now edit the 'x' (Private)

and ‘y’ (DMZ) values for your network:

```
set interfaces ethernet eth0 address dhcp
set interfaces ethernet eth1 address 10.10.x.254/24
set interfaces ethernet eth2 address 10.10.y.254/24
set protocols static route 0.0.0.0/0 next-hop 10.221.3.254

set nat source rule 1 outbound-interface eth0
set nat source rule 1 source address 10.10.x.0/24
set nat source rule 1 translation address masquerade

set nat source rule 2 outbound-interface eth0
set nat source rule 2 source address 10.10.y.0/24
set nat source rule 2 translation address masquerade
```

Now, from Kali, create an SSH connection to the default gateway on the firewall (10.10.x.254):

```
ssh 10.10.y.254 -l vyos
```

Now log into the firewall, and go into configuration mode and copy-and-paste the config from your config file, check the config, and then commit the changes if they are correct:

```
Login:
Password: ******
$ configure
# <paste-config>
# commit
```

Now setup your Ubuntu host with 10.10.x.7 with a default gateway of 10.10.x.254, and the Windows 7 host with 10.10.y.7 with a default gateway of 10.10.y.254. Note, on Kali and Ubuntu, you need to edit /etc/resolv.conf and add a nameserver of 10.221.3.254.

From Kali, can you ping the local host, the Ubuntu host, the Windows host, the firewall ports, 8.8.8.8 and google.com? [Yes][No]

From Windows, can you ping the local host, the Ubuntu host, the Kali host, the Windows host, the firewall ports, 8.8.8.8 and google.com? [Yes][No]

From Ubuntu, Kali and Windows, can you access google.com from a browser [Yes][No]

Is everything working on your network? [Yes][No]

Now nmap from the Ubuntu host to the Windows host. Which ports are accessible:

[ftp, ssh, http, https]

Now nmap from the Windows host to the Ubuntu host. Which ports are accessible:

[ftp, ssh, http, https, rpcbind, vnc]

D Setting up Firewall Rules

Now we will setup firewall rules between **zones** (networks) connected to the firewall. In this case we will enable all the connections from the private network to the DMZ, but only allow TCP ports

80 and 443 to go through from the DMZ to the private network. All other connections will be disallowed. If we allow the connections from the private and the DMZ, we must remember the connection to allow it back from the DMZ to the private network, thus we define that we accepted established connections.

Now we will configure the next part of the firewall by copying-and-pasting the config to the firewall. First download the following config:

`https://asecuritysite.com/vpart02.txt`

```
set zone-policy  zone  private  description  "Inside"
set zone-policy  zone  public  description  "Outside"
set zone-policy  zone  dmz description  "DMZ"

set zone-policy  zone  public  interface  eth0
set zone-policy  zone  private interface  eth1
set zone-policy  zone  dmz  interface  eth2

set  firewall  name  private2dmz description  "private to DMZ"
set  firewall  name  private2dmz rule  1  action  accept
set  firewall  name  private2dmz rule  1  state  established  enable
set  firewall  name  private2dmz rule  1  state  related enable
set  firewall  name  private2dmz rule  10  action  accept
set  firewall  name  private2dmz rule  10  destination  port  80,443
set  firewall  name  private2dmz rule  10  protocol tcp

set  firewall  name  dmz2private description  "DMZ to private"
set  firewall  name  dmz2private rule  1  action  accept
set  firewall  name  dmz2private rule  1  state  established  enable
set  firewall  name  dmz2private rule  1  state  related enable

set  zone-policy  zone  private from  dmz firewall  name  dmz2private
set  zone-policy  zone  dmz from  private firewall  name  private2dmz
```

and paste it into your firewall, check the config, and then **commit**. This will block all the connections, apart from ones to TCP Port 80 (http) and TCP Port 443 (https) from the private network to the DMZ.

From Ubuntu, can you ping the local network, the Windows host, the firewall ports and 8.8.8.8. Can you now ping?

[Yes/No]

From Ubuntu, can you access the Web server on Windows from a browser [Yes/No]

From Windows, can you ping the local network, the Windows host, the firewall ports and 8.8.8.8. Can you now ping?

[Yes/No]

From Windows, can you access the Web server on Ubuntu from a browser [Yes/No]

From Ubuntu and Windows, can you access Google.com from a browser [Yes/No]

Now nmap from the Ubuntu host to the Windows host. Which ports are accessible:

[http, https]

Now nmap from the Windows host to the Ubuntu host. Which ports are accessible:

[]

Explain the operation of the network with the new network settings:

E Allowing access to the public network

You should not currently be able to connect from the private network to the public one. Now setup this connection:

http://asecuritysite.com/vpart03.txt

```
set firewall      name  private2public description    "private to public"
set firewall      name  private2public rule     1     action      accept
set zone-policy   zone  public from private firewall  name  private2public

set firewall      name  public2private description    "public to private"
set firewall      name  public2private rule     1     action      accept
set firewall      name  public2private rule     1     state established enable
set firewall      name  public2private rule     1     state related enable
set zone-policy   zone  private from public firewall  name  public2private

commit
```

You should now be able to connect from the private network to the public one.

From Ubuntu, can you access 8.8.8.8 with ping? [Yes/No]
From Ubuntu, can you access google.com with ping? [Yes/No]
From Ubuntu, can you access google.com from a browser? [Yes/No]

From Windows, can you access 8.8.8.8 with ping? [Yes/No]
From Windows, can you access google.com with ping? [Yes/No]
From Windows, can you access google.com from a browser? [Yes/No]

Now create your own config and allow the DMZ to communicate with the public network.

You should now be able to connect from the private network to the public one.

From Ubuntu, can you access 8.8.8.8 with ping? [Yes/No]
From Ubuntu, can you access google.com with ping? [Yes/No]
From Ubuntu, can you access google.com from a browser? [Yes/No]

From Windows, can you access 8.8.8.8 with ping? [Yes/No]
From Windows, can you access google.com with ping? [Yes/No]
From Windows, can you access google.com from a browser? [Yes/No]

Which configuration commands have you used:

Can you connect your Windows host to the Google.com? [Yes][No]
Can you connect your Ubuntu host to the Google.com? [Yes][No]

F Snort IDS

If this part, you will need open up your firewall. If you want to quickly do this, you can run:

```
$ configuration
# delete zone-policy
# commit
```

Make sure you Ubuntu and Windows hosts can connect to each other, and to the Internet.

F.1 Snort on Ubuntu

On Ubuntu, create simple Snort rules files both called **mysnort.rules**, and add the following rules:

```
alert tcp any any -> any 443 (sid:999; msg:"Port 443")
alert tcp any any -> any 80 (sid:1000; msg:"Port 80")
```

The format of Snort Detection Rules are as follows:

```
action protocol src-ip src-port > dest-ip dest-port (packet-payload-params output-msg)

[pass|log|alert] [ip|icmp|tcp|udp] [any|IP] [any|port] > [any|IP] [any|port]
([content:"searchstring";], [nocase;], [msg:"alert message";] sid:ruleid;)
```

From the Ubuntu, run Snort (used ifconfig to see your interfaces – and you will need to create a folder named log):

```
snort -c mysnort.rules -i ens32 -p -l log -K ascii -k none
```

From Ubuntu, start Snort with the rules to detect access to Port 443 and Port 80. Now access www.google.com, and then stop Snort and examine the log.

Did Snort detect the connection? [Yes/No]

What information is contained in the Snort log:

From Windows, start Snort with the rules to detect access to Port 443 and Port 80. Now access www.google.com, and then stop Snort and examine the log.

Did Snort detect the connection? [Yes/No]

What information is contained in the Snort log:

Now we will detect the word "bbc" in the traffic for DNS access. In the Snort rules file, add another rule

of:

```
alert udp any any -> any 53 (sid:1001; content:"bbc"; nocase; msg:"DNS call
for bbc")
```

Now run Wireshark on Ubuntu and capture traffic. Then run Snort.

Access bbc.com from the browser on Ubuntu.

Perform a DNS looking using "nslookup bbc.co.uk".

Now stop Snort and Wireshark. Now examine the alert file.

Did it detect each of the accesses? [Yes/No]

Now examine the Wireshark trace.

Can you find the network packages related to the DNS access? [Yes/No] (you may have to filter with "udp.port==53")

F.2 Snort on Windows

Now go to Windows, and run Snort from the required network interface:

```
snort -c mysnort.rules -i 2 -p -l log -K ascii -k none
```

Now access www.google.com, and then stop Snort and examine the log.

Did Snort detect the connection? [Yes/No]

What information is contained in the Snort log:

From Windows, start Snort with the rules to detect access to Port 443 and Port 80. Now access www.google.com, and then stop Snort and examine the log.

Did Snort detect the connection? [Yes/No]

What information is contained in the Snort log:

Now we will detect the word "bbc" in the traffic for DNS access. In the Snort rules file, add another rule of:

```
alert udp any any -> any 80 (sid:1001; content"bbc"; nocase; msg:"DNS call for
bbc")
```

Now run Wireshark on Ubuntu and capture traffic. Then run Snort.

Access bbc.com from the browser on Windows.

Perform a DNS looking using “nslookup bbc.co.uk”.

Now stop Snort and Wireshark. Now examine the alert file.

Did it detect each of the accesses? [Yes/No]

Now examine the Wireshark trace.

Did it detect each of the accesses? [Yes/No] (you may have to filter with “udp.port==53”)

F3 Detecting a word in a network packet

On Windows, go to the c:\inetpub\wwwroot folder, and then edit the iisstart.htm file, and add the HTML code of:

```
<h1>Computer Security and Cryptography</h2>
<p>This is our home page of the Napier module on IIS</p>
```

Save the file, and then access the page from Ubuntu. Now, create a rule on Ubuntu to detect the word “module” within a network connection.

Did Snort detect the word “module”? [Yes/No]

Which Snort rule did you use:

On Ubuntu, go to the \var\www\html folder, and then edit the index.html file, and add the HTML code of:

```
<h1>Computer Security and Cryptography</h2>
<p>This is our home page of the Napier module on Apache</p>
```

Save the file, and then access the page from Windows. Now, create a rule on Windows to detect the word “module” within a network connection.

Did Snort detect the word “module”? [Yes/No]

Which Snort rule did you use:

F4 Detecting HTTPs

On Ubuntu, now add a rule to detect the word “google” in an HTTPs connection:

```
alert tcp any any -> any 443 (sid:1002; msg: content: "google"; "Port 443")
```

Now test the rule.

Did Snort detect the word “google”? [Yes/No]

Assuming that the word “google” was in the data packets, why was it unable to find the word?

G Software Tutorial

Complete the software tutorial at:

https://github.com/billbuchanan/csn09112/tree/master/week03_ns/labs/additional_lab

Appendix

```
configure

set interfaces ethernet eth0 address dhcp
set interfaces ethernet eth1 address 10.10.x.254/24
set interfaces ethernet eth2 address 10.10.y.254/24
set system gateway 10.221.3.254

set nat source rule 1 outbound-interface eth0
set nat source rule 1 source address 10.10.x.0/24
set nat source rule 1 translation address masquerade

set nat source rule 2 outbound-interface eth0
set nat source rule 2 source address 10.10.y.0/24
set nat source rule 2 translation address masquerade

set zone-policy zone private  description  "Inside" set zone-policy zone
public  description  "Outside" set zone-policy  zone  dmz description "DMZ"

set zone-policy zone public  interface  eth0
set zone-policy zone private interface  eth1
set zone-policy  zone  dmz  interface eth2

set firewall      name  dmz2private description "DMZ to private"
set firewall      name  dmz2private rule  1     action      accept
set firewall      name  dmz2private rule  1     state established enable
set firewall      name  dmz2private rule  1     state related enable
set firewall      name  dmz2private rule  10    action accept
set firewall      name  dmz2private rule  10    destination port  80,443
set firewall      name  dmz2private rule  10    protocol tcp

set firewall      name  private2dmz description "private to DMZ"
set firewall      name  private2dmz rule  1     action      accept

set zone-policy   zone  private from      dmz firewall      name  dmz2private
set zone-policy   zone  dmz from    private firewall  name  private2dmz


set firewall      name  private2public description    "private to public"
set firewall      name  private2public rule     1     action      accept
set zone-policy   zone  public from  private firewall  name private2public

set firewall      name  public2private description    "public to private"
set firewall      name  public2private rule     1     action      accept
set firewall      name  public2private rule     1     state established enable
set firewall      name  public2private rule     1     state related enable
set zone-policy   zone  private from  public firewall  name public2private
```

http://asecuritysite.com/vfinal.txt